SCIENTIFIC PROGRAMME

SIMULATION METHODOLOGY

A FORMAL MODEL FOR THE SEQUENTIAL, UN-TIMED SUBSET OF SYSTEMC

Primrose Mbanefo and Wolfgang Raab
Infineon Technologies AG
Am Campeon 1-12, Neubiberg,
85579, Germany
{primrose.mbanefo,wolfgang.raab}@infineon.com

Pierre Wodey
ISIMA/LIMOS
BP 10125, 63173, Aubière Cedex,
France
pierre.wodey@isima.fr

KEYWORDS
Formal model, sequential, un-timed, SystemC.

ABSTRACT

SystemC is a system level design language based on C++. There are a number of situations, like the case of formal verification, which necessitate a formal model of a SystemC program. This paper investigates the extraction of a formal model for theorem proving purposes from a subset of SystemC. A suitable formal model is chosen as well as described. The rules of translation from SystemC to the model are given. The setup is tested on two simple examples. Functional properties of the examples were proven.

INTRODUCTION

There is a rise in the demand for embedded information processing systems. These systems have to be small enough to fit in the target environment but should also be just as functionally efficient as personal computers. This trend can be seen in automobiles and in smart phones. The smallest possible space is obtained if the entire system can be put on a single chip. The design of such a chip needs to be capable of handling both the hardware and software parts of the chip, give an overview of the system although the implementation has not yet been decided on, as well as permit architecture exploration. Industry is currently turning to languages capable of the above and one of these is SystemC.

The ultimate goal of the research done for this paper is the proof of correctness of a full System-on-Chip (SoC) in SystemC using theorem proving methods. Other attempts at this are (Akbarpour and Tahar 2003), with verification emphasis on fixed point arithmetic, and (Kalla et al. 2005), which involves the extraction of a formal model used for multiple purposes including theorem proving. Our goal has been divided into four problems to allow an incremental investigation of the possibilities. These are the proof of correctness of:

- the sequential, un-timed blocks: These are monolithic blocks. This resumes to investigating the individual processes without any events or to the formal verification of a C program. It is the focus of this paper.
- concurrency: The study of this problem shows how best to deal with the interaction of the different entities. This interaction will be considered on the modular level where all modules are started simultaneously. It will also be studied on the process level where each string of instructions between two events can be considered as a monolithic block.
- timing: This would show how to reason about the parts of SoCs which depend on clocks.
- typing: This problem tackles the complexity introduced by typing in software engineering. Unlike types in hardware which are geared towards physical machines, software uses pointers, casts, overloading and polymorphisms among others, obscuring the type space.

This paper looks at the extraction of a formal model from the monolithic SystemC subset and proof of its correctness. Such extraction is traditionally done manually, but the target verification environment found is expressive enough to permit compilation towards it, automating at least one part of the process. The formal verification of monolithic programs is in itself nothing new, but needs to be done for SystemC. The work done for this paper involved choosing a suitable formal model, compiling towards it and validating the procedure on simple tests.

Section 2 discusses the need for a formal model of SystemC programs. An overview of the formal model chosen is given in section 3. Section 4 shows the rules used in translating from SystemC to the formal language. The examples the verification environment was tested on are overviewed in section 5 and section 6 discusses the conclusions and future directions.

NEED FOR A FORMAL MODEL

SystemC (Open SystemC Initiative 2002) is a C++ library which provides a syntax and semantics for SoC modeling. It also provides a process scheduler. This imperative, executable form of the language is well adapted to functional evaluation by simulation. Models in SystemC are executable specifications. How much time is spent in a functional block of the system or how many times a particular function is used are examples of information which can be retrieved during the evaluation process. A formal model is reliant on a mathematical description of the system. It is therefore better adapted to verification, automatic refinement and synthesis of the system since there is only a formal evolution and evaluation of the system. Proof of system correctness increases confidence in analysis done during simulation.

THE CHOSEN FORMAL MODEL

Formal model candidates are, for this subset, models geared towards representing imperative programs. Some are state machine representations, lambda-calculus and the Hoare logic (Clarke and Wing 1996). Each has its advantages and disadvantages but a Hoare logic oriented model was chosen over others because of the ease of SystemC translation towards it, its integration into the proving environment, Isabelle/HOL (Nipkow et al. 2002), and its readability. The readability provides a means for verifiers and designers to discuss about the system.This model developed by (Schirmer 2005) is given as a language with formal semantics. The basic language constructs are:

Skip : Do nothing
Basic f : Basic commands like assignment
Seq c1 c2: Sequential composition, also written as c1;c2
Cond b c1 c2: Conditional statements
Guard g c: Guarded commands, also written as g ↦ c
While b c: for looping
Call p: Static procedure calls
Throw : Initiate abrupt termination
Catch c1 c2: Handle abrupt termination of c1 with c2
DynCom c: Dynamic (state dependent) command. This command is used to implement side-effect expressions, pointers to procedures, and dynamic method invocation among others. Unlike the other commands, it does not work on the current state but is given a state to act on.

The language model is formal in that its semantics are stated formally using an operational big-step approach. It is Hoare logic based as it comes with a verification environment which describes a Hoare logic for both the partial and total correctness of programs in the language.

THE SUBSET AND TRANSLATION RULES

This section only describes the part of SystemC necessary for this paper. We also look at how we derive the formal model from the SystemC subset. The rules of translation used in this section are of the form: $\dfrac{Conds}{C \text{ ` } SC \rightarrow FM}$ where $Conds$ is the set of conditions which, when all true, validate the translation rule, $C \text{ ` } SC \rightarrow FM$. C is the context in which SC, the SystemC entity, is being translated. It contains the variables used by the entity and their types. *FM* is the formal model, the result of translation. It is the program in the formal language derived from the SystemC entity.

SystemC main function

The first element in this subset is the SystemC main function which tells the SystemC kernel which modules are instantiated in the program. Its code snippet is:

```
int sc_main(int argc, char** argv) {
Module instance("name");
sc_start(-1); /*timing is ignored.*/
return 0; }
```

With focus on a non-concurrent subset, there can be only one module with one instance. The subset is also reduced to untimed programs, there is, to this effect, no instance of a clock. The following rule can therefore be used without any loss of information:

$$\frac{}{\text{ ` MAIN} \rightarrow \begin{array}{l} \textit{theory}\ \text{sc\_main}\ \textit{imports}\ \text{instance}\ \textit{begin} \\ \textit{procedures}\ \text{main} = \text{"}\textit{CALL}\ \text{module()"} \\ \textit{end} \end{array}}$$

This states that code representing the sc_main function can unconditionally be translated into a theory which uses the formal model derived from the module instance. The sc_main function becomes a procedure which calls the procedure representing the module instance functionality.

SystemC modules

Given the C++ basis of SystemC, there are multiple acceptable ways of declaring a module. Here is one of them.

```
class Module : public sc_module {};
```

Modules can have members i.e. data, and processes i.e. functionality. Module members are at the moment restricted to data with SystemC and C++ data types without pointers. Pointers and user defined data types, belong to the previously mentioned complexity introduced by software engineering. We can only have one process because the behaviour of multiple processes in a module is concurrent. Let this process be *Proc* and its local variables be found in the context *l*. Let the context for the module be $g \lhd m$, where the operator $\lhd$ is used to pile contexts. In context, C1 $\lhd$ C2, variables are first of all searched for in C2 and then in C1. *g* contains global variables and *m* contains module members. The context used by *Proc* is $g \lhd m \lhd l$. This context is translated into the memory model of the formal model which we will call *memory*. The following rule states that: if the procedure is translated to FM_{PROC} using its context then the formal model of the module uses *memory* to first of all produce FM_{PROC} . The module functionality is then represented by a procedure which calls the procedure representing the process functionality.

$$\frac{g \lhd m \lhd l\text{` Proc} \rightarrow FM_{PROC}}{\text{ ` MODULE} \rightarrow \begin{array}{l} \textit{theory}\ \text{module}\ \textit{imports}\ \text{memory} \\ \textit{begin}\ FM_{PROC} \\ \textit{procedures}\ \text{module} = \\ \text{"}\textit{CALL}\ \text{Proc()"}\ \textit{end} \end{array}}$$

SystemC processes

The process declaration can also be done in many ways. One of which is :

```
SC_HAS_PROCESS(ModuleName);
Module(sc_module_name name):
sc_module(name){SC_METHOD(ProcessName);}
```

SystemC provides 2 types of processes. The SC_METHOD and the SC_THREAD. This subset concentrates only on the SC_METHOD, a process executed in its entirety each time it is executed. It is at the moment, the only process running.

The SC_THREAD is as the name implies a thread of execution which can be stopped at any point in the execution and continued by its wait statements and events. It will be used when concurrency will be studied.

A process can, at the moment, contain anything a C++ function can contain except function calls. A process, Proc, is translated by:

$$\frac{g \lhd m \lhd l^{\backprime}\ \text{Statement} \rightarrow FM_{Statements}}{g \lhd m \lhd l^{\backprime}\ \text{PROCESS} \rightarrow \begin{array}{l} \mathit{procedures}\ \text{Proc} = \\ "FM_{Statements}" \end{array}}$$

There is a translation rule for each possible statement of a process but they are not all here for space and clarity. Here is an example of the "do statement". It states that given the translations for the condition and the body of the statement, we can execute the body once and then execute it as long as the condition is valid.

$$\frac{\begin{array}{c} g \lhd m \lhd l^{\backprime}\ \text{DO\_COND} \rightarrow FM_{DO\_COND} \\ g \lhd m \lhd l^{\backprime}\ \text{DO\_BODY} \rightarrow FM_{DO\_BODY} \end{array}}{\begin{array}{l} \qquad\qquad\qquad\qquad\qquad\quad FM_{DO\_BODY} \\ g \lhd m \lhd l^{\backprime}\ \text{DO\_STMT} \rightarrow \mathit{WHILE}\,(FM_{DO\_COND}) \\ \qquad\qquad\qquad\qquad\qquad\quad \mathit{DO}\ FM_{DO\_BODY}\ \mathit{OD} \end{array}}$$

APPLICATIONS

A compiler was built in order to automatically derive a program in the formal language model from a SystemC program based on the above rules. It was then used on two simple examples, bubble sort and a FIR filter to generate formal representation for them. These models as well as the properties which we wished to prove on them were then loaded into Isabelle/HOL.

The SystemC bubble sort program, a software oriented, data centric example, is 22 lines of code in a single file. 3 files with a total of 46 lines of code for the formal model were generated. The proof of lists being sorted is 50 steps long containing 1 auxiliary lemma. Thinking up the steps takes its time but their execution is done in a couple of seconds. The same holds for the Fir filter, a hardware oriented, calculation centric, SystemC program. It is a modified version of the Fir filter provided as an example in the SystemC directory. The SystemC program is 46 lines of code separated over 4 files. It is compiled into 3 files with a total of 61 lines of code. The proof of functionality is 183 proof steps long comprising 6 auxiliary lemmas.

CONCLUSION

There is a need for a formal modeling of SystemC programs. Since this is work in progress, this paper covered the formal modeling of one of the simple parts of SystemC, the monolithic subset. The interest of such a setting is to verify the core functionality of SystemC designs i.e. the individual processes before we go further to verify the structure and protocols of the chip design in the concurrent subset. Working on this led to the choice of a formal model, rules for translation to this model and the development of a compiler from SystemC to the formal language model. The setup was tested on two examples namely, bubble sort and a FIR filter written in SystemC. Their functionality was formally verified. The next step would entail looking at how to model the concurrent parts of the system. The first step would be identifying which theory on concurrency modeling would be best adapted to our needs. Experiments are currently being done with UNITY (Paulson 2001), a Hoare logic for parallel programs (Nieto 2001), the Pi-Calculus (Milner et al. 1992a,b), CSP (Hoare 1978) and TLA+ (Lamport 2002). The next step would be finding out the atomic instructions in a SystemC program. The current idea is to analyze a process and take the part between two wait statements to be atomic since it will execute to its end without interruption. Cutting up the program in this way will surely affect readability, making the third step finding out how we can improve readability of the extracted formal model.

ACKNOWLEDGMENTS

This work was partially funded by the German Federal Ministry of Education, Science, Research and Technology (BMBF) in the framework of the Verisoft project (www.verisoft.de) under grant 01 IS C38. The responsibility for this article lies with the authors.

REFERENCES

Akbarpour, B. and S. Tahar: 2003, 'Modeling SystemC Fixed-Point Arithmetic in HOL'. *Lecture Notes in Computer Science 2885*, 206–225.

Clarke, Jr., E. M. and J. M. Wing: 1996, 'Formal methods: State of the art and future directions'. *ACM Computing Surveys* 28(4), 626–643.

Hoare, C. A. R.: 1978, 'Communicating Sequential Processes'. *Communications of the ACM* 21(8), 666–677.

Kalla, H., D. Berner, J.-P. Talpin, and L. Besnard: 2005, 'A methodology to automatic building of formal models from SystemC description'. Rapport d'activité 66, INRIA, Project team : espresso.

Lamport, L.: 2002, Specifying Systems: *The TLA+ Language and Tools for Hardware and Software Engineers*. Pearson Education, Inc, 1st edition.

Milner, R., J. Parrow, and D. Walker: 1992a, 'A Calculus of Mobile Processes 1'. Information *and Computation* 100(1), 1–40.

Milner, R., J. Parrow, and D. Walker: 1992b, 'A Calculus of Mobile Processes 2'. *Information and Computation* 100(1), 41–77.

Nieto, P. L.: 2001, 'Verification of Parallel Programs with the Owicki-Gries and Rely-Guarantee Methods in Isabelle/HOL'. Phd thesis, Technische Universität München.

Nipkow, T., L. C. Paulson, and M. Wenzel: 2002, *Isabelle-HOL A Proof Assistant for Higher-Order Logic*, Vol. 2283 of *Lecture Notes in Computer Science*. Springer.

Open SystemC Initiative: 2002, 'SystemC User's Guide'. PDF available at systemc.org.

Paulson, L. C.: 2001, 'Mechanizing a theory of program composition for UNITY'. *ACM Transactions on Programming Languages and Systems* 23(5), 626–656.

Schirmer, N.: 2005, 'A verification environment for sequential imperative programs in Isabelle/HOL'. *Logic for Programming, Artificial Intelligence, and Reasoning*, *Proceedings* 3452, 398–414. Lecture Notes in Computer Science.

SIMULATION VALIDITY ASSESSMENT TAILORING WITH UML

V. ALBERT, A. NKETSA and M. PALUDETTO
LAAS-CNRS
314 avenue du Colonel Roche
31500 Toulouse
France
E-mails: {valbert,alex,mario}@laas.fr

KEYWORDS
UML, VV&A, M&S, System Engineering, Quality Assurance

ABSTRACT

Broadly speaking, Modelling and Simulation (M&S) has become a common activity in order to predict, reproduce and analyse a dynamic system behaviour providing an effective way for engineering decision making as well as to gather a larger knowledge of this system. The prerequisite of M&S activities is to determine how well the M&S product reflects the relevant system behaviour and build a full confidence in M&S products. The process of determining this is referred to as assessment of simulation products validity. Simulation validity assessment is a wide research domain that interests large number of scientific communities and industries. While scientific communities have focused over terminologies, methods and processes since many years, it has not matured anymore in term of techniques and tools due to the complexity and the diversity of the scientific domain research involved. Through this contribution, we would like to focus on existing standards and tools, especially UML, in order to provide a uniform base of documentation and exchange of Verification, Validation and Accreditation (VV&A) information and concepts. The motivation for such a study was to lighten and fix all items involved in M&S VV&A and their relationships and therefore build a solid foundation for further study.

INTRODUCTION

At Airbus Industry there is a steadily increasing tendency to use simulation to validate systems requirements that are too complex to be confidently validated by examining alone. As a matter of fact, this is a natural and desirable evolution from the wide use of simulation during the development of new aircraft. Nowadays, Airbus is highlighting the question of models and simulation results validity for embedded systems. Also they wish to introduce in their current methods and processes a VV&A framework but differently of current practices (e.g. US DoD). The non-negligible difference with defense domain for instance, that let us suppose to have success in this study, is the fact that the simulation world is closely time ahead of the real world and both are evolving in parallel. In that way as M&S is used, relatively rich real systems knowledge is available. In that sense, the system of reference, in many cases, would be the real world itself.

This paper aims to clarify and fix VV&A and M&S elements and *their relationships*. This study would serve as a solid foundation to build up VV&A processes within M&S activities. It would also encourage adherence to the right approach and good practices. While M&S and VV&A involves a wide variety of aspects of different nature making simulation validation assessment a difficult task this paper focuses on some of the general concepts of M&S and VV&A. As communities are not exploiting existing technology as much as desired and have not given adequate attention to the benefits of tools and technologies, this paper introduces a UML based framework that may be used for VV&A of models and simulation results.

VV&A and M&S communities have to keep in their mind that we should identify existing techniques to support such process coming from different research and development areas, including *requirements engineering*, *system analysis in the application domain*, *software engineering and software quality assurance.* Thus first part will introduce M&S elements and processes. We are addressing a methodological approach derived from system engineering standards. The second part introduces VV&A concepts and relates them to software engineering concepts. In a third part we set out our framework and illustrate it through an example.

MODELLING AND SIMULATION

To introduce the concepts of M&S we used the framework for Modeling and Simulation (Zeigler, Praehofer and Kim 2000). This framework defines the central entities of an M&S products and their relationships (figure 1).

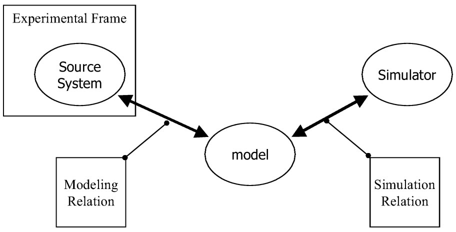


Figure 1. M&S Framework

The System

The system is the real or virtual system which is used as the source of observable data. In (METHGU2 2004), it is called the *system of interest* that is a set of fictive or existing entities and their interactions **subjected to modelling and simulation.** This definition highlights the difference between *real world* and *real system* or *system*. Indeed in (METHGU2 2004) the system, called the real system consists of a cut out of the real world with a well-defined system border. The system border, which will be defined through the objectives of the study, is the limit between the system of interest and the rest of the real world (left side of figure 2 illustrates this concept).

The Experimental Frame

The *experimental frame* specifies the conditions under which the system is observed or experimented. It is an operational formulation of the objectives and needs of an M&S application. In others words the experimental frame aims to traduce objectives in precise experimentation conditions (constraints, required interfaces, contents of the real system substitute…). We can see it as a system that interacts with the system of interest in order to obtain the required data under specific conditions.

The Model

The *model* is the representation (substitute) of the system within a specific experimental frame of interest. It is typically a set of instructions, rules, equations or constraints to generate behaviour.

For modelling a system three different representation forms of a model can be taken into considerations: (Brade 2004)

- The *Conceptual Model (CM)* describes the abstracted and idealized representation of the real system and holds all concepts of the model or the simulation.

- The *Formal Model (FM)* is the formalized description of the Conceptual Model, compliant with a well-defined modeling formalism, expresses the Conceptual Model quantitatively and unambiguously.

- The *Executable Model (EM)* technically implements the Formal Model and provides the additional information that allows the model to be executed and operated on a computer.

The following figure tends to illustrate the whole concepts identified above and their dependences - the gap between reality and system of interest and the gap between real world and simulation world.

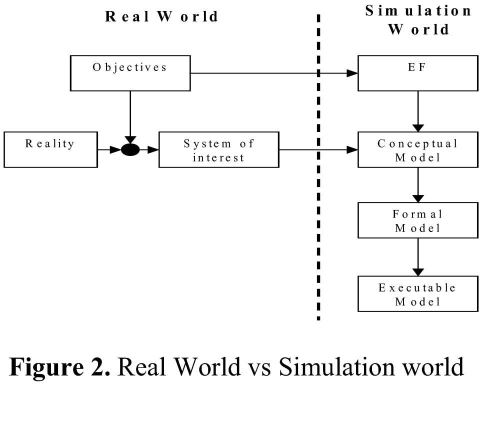


Figure 2. Real World vs Simulation world

The Simulator

In the M&S framework the simulator is a computation system allowing to execute the model and to generate its behaviour from model's instructions.

Simulation

If an analytical solution of the model is not feasible due to the limitations of the modeling formalism or model complexity, one may approximate the behavior of the real system by executing the model over time, and subsequently or interactively draw conclusions about reality from the observed dynamic behavior of the model.
In the context of our study, *simulation* means experimentation with a model. A *simulation run* is a single execution of the model, yielding *simulation results* or model output. The execution of a *simulation experiment* consists of a set of *simulation runs*, which must be well planned, and requires an *experiment design* for the model similar to the setup of a real experiment. If the experiment design is unsuitable, results from simulation may not allow the desired increase in knowledge about the real world, or may be statistically insignificant. There are numerous distinct simulation methods and techniques available. The selection of a particular simulation method depends on the type of the model and the intended use of the simulation results. Here, simulation is defined in analogy to (IEEE 610.3 1989).

Modelling and model development process

All the system and software engineering methodologies define some products and processes associated with documentation, configuration management, quality assurance, verification and validation (for software or computer hardware), etc… M&S VV&A shall not redefine these methodologies but the VV&A methodology must identify, in all these activities, the M&S specific requirements, when they exist or when they have a specific importance, and support them (METHGU2 2004).

Model Development within the Systems Engineering Process
The Model development is a centric process of the Systems Engineering. It was based a long time on good engineering practice and experience. As systems became more complex to include software and human interactions, engineering disciplines and organizations often became fragmented and specialized in their attempt to cope with this increasing complexity. Organizations focused on the optimization of their primary products and often lost sight of the overall system. Each organization perceived that their part must be optimal, using their own disciplinary criteria, and failed to recognize that all parts of a system do not have to be optimal for the system to simulate and perform optimally. *The need to recognize that system requirements can differ from disciplinary requirements is a constant challenge in systems development* (Incose 2004), and in model development as well**.** The Systems Engineering process, especially the modelling process, should be viewed as a major effort in communication and the management of teams of experts that lack a common paradigm and a common language but must work together to achieve a common goal.

The Need of Methodological Approach

The basic engine for the model development of a system is an iterative process that expands on the common sense strategy of model a little, verify a little and test a little. The iteration includes steps to (1) understand a problem before we attempt to model it, (2) create alternative solutions (3) examine this solutions with respect to the objectives of the modelling (design, performance evaluation, simulations), and (4) verify that the selected solution is correct before continuing the model development activities or proceeding the next step.

In a generic approach, the basic Model development process tasks are:

(1) Define the System Objectives (User's Needs)
(2) Establish the Functionality (Functional Analysis)
(3) Establish Performance Requirements (Requirements Analysis) for design, simulation, or other objective)
(4) Define the baseline modelling and simulation (e.g. Model Driven Architecture[1])
(5) Define or analyse the target platform (specific hardware and languages)
(6) Evolve Design/Simulation and Operations Concepts (Architecture Synthesis)
(7) Establish basic models according the baseline selected and Requirements (design and/or simulation objectives, PIMs models whether MDA is chosen)
(8) Verify the basic models (formal whether formal languages are used)
(9) Select the better model according the objectives to meet
(10) Transform it into target hard and soft Requirements (PSM for MDA approach)
(11) Verify that the Baseline Model Meets Requirements: Verify and Validate the resulting specific platform Model (formal and simulation techniques whether necessary)
(12) Validate that the Baseline Model Satisfies the User (User's Needs)
(13) Iterate the Process through Lower Level Model Analysis (Decomposition)

These tasks are implemented through the process shown in Figure 3 for the System Simulation, derivate from the EIA-632 standard reference process of systems engineering. The basic tasks listed above are performed in the Model Simulation Design process block. The process involves a Requirements Definition Process that establishes both performance (quantified) requirements and functional area (architecture) requirements, and a Solution Definition Process that translates those requirements into design/simulation and operational model solutions. Overarching Technical Management Processes and Technical Evaluation Processes are performed continuously throughout the Simulation model development processes.

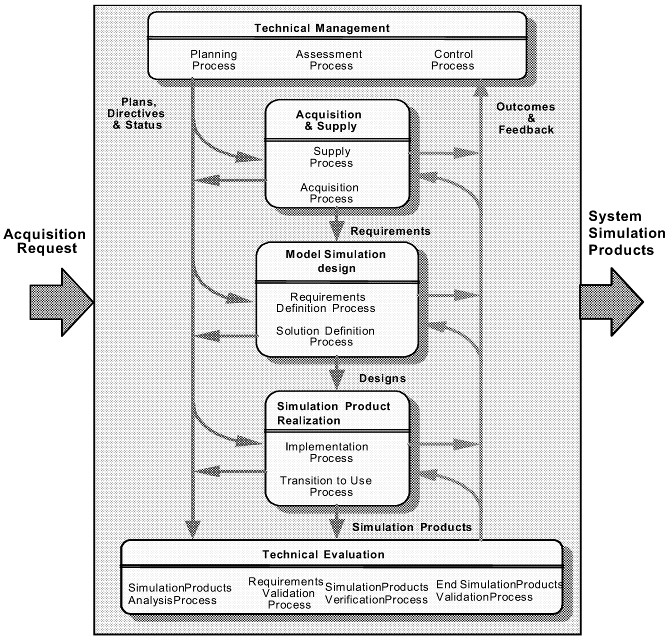

Technical Management
Planning Process
Assessment Process
Control Process
Plans, Directives & Status
Outcomes & Feedback
Acquisition & Supply
Supply Process
Acquisition Process
Requirements
Acquisition Request
System Simulation Products
Model Simulation design
Requirements Definition Process
Solution Definition Process
Designs
Simulation Product Realization
Implementation Process
Transition to Use Process
Simulation Products
Technical Evaluation
SimulationProducts AnalysisProcess
Requirements Validation Process
SimulationProducts VerificationProcess
End SimulationProducts ValidationProcess

Figure 3. System Simulation Products Process Overview **(derived from ANSI/EIA-632)**

The Systems Engineering process of the model simulation is used iteratively in the Model development cycle to generate more detailed descriptions of the simulation models. These models are created, documented and maintained within classes in a Database connected to the simulation platform. The simulation models are stored on the Database as a set of classes structured in a class hierarchy. Each class includes *what* needs or part of needs to be achieved (functional requirements and concept of operations attached to it), *how well* it must be achieved (performance requirements and technical performance measures), *how* it is to be achieved (model design), *how* it is linked to the rest of the needs (traceability) and the results of the analysis and tests that gives the ability to satisfy requirements (verification) user needs (validation).

The Modelling process leads to the evolution of a more detailed database classes for each subsequent phase in the development cycle (through the final/critical model review in a full-scale modelling program). Each phase starts from an upper level of database classes of the system and first decomposes and assigns the upper level functional descriptions, and then the upper level requirements into lower level requirements for each child function within class objects. The decomposed objects/function/requirement sets are then documented into specifications. Trade off studies can then be carried out to search for the most cost/beneficial solutions, and the capability verified through test and analysis. Their models can be enhanced towards the iterations and the successive simulations.

[1] The Model Driven Architecture "MDA" (OMG site) is increasingly used to define an approach to system development based on modeling, and automated mapping of models to implementations. It becomes even so for the production of simulation modeling too, where several domains and trades are strongly involved. It recent industrial success, along with it use in an ever greater number of industrial projects prompt engineers and researchers to define and use MDA based approaches. The basic MDA concept involves defining a Platform Independent Model "PIM" and its automated mapping to one or more Platform-Specific models "PSM" from the Platform Dependent Model too. The virtual prototyping designed with an MDA view allows you to make a system V&V (Validation and Verification) by means of both, the formal verification and the system simulation.

VERIFICATION, VALIDATION AND DECISION

Verification and Validation (V&V) are techniques and actions applied to a product and its intermediate products, issued from the different development process phases in order to ensure they conform to pre-establish dispositions.
V&V has been widely accepted in M&S communities while it is also used in related field, e.g., in systems engineering (however we will see that validation in M&S domain differs from system engineering). M&S communities have introduced another concept in order to convey the aspect of confidence in simulation results. This concept is used to be defined as Accreditation (DMSO 2000), Credibility (Brade 2004) or Acceptance (METHGU2 2004). Anyway how we call it, in our study it is seen as the final decision that will be taken to say whether or not a specific M&S can get something out to the user and if he can use it. In that way, V&V aims to provide *evidence* and *indication* in order to take this final decision.

Verification and correctness

In system engineering, verification is the activity that consists to make sure that an intermediate product (or final stem) from a specific development cycle phase is conforming to the referential of the previous phase. In M&S context, verification deals with the examination of correctness. The term of correctness is also used in software verification. (METHGU2 2004) definition for Verification illustrates well this idea: *Verification: The process, which is used to construct, under a set of time, cost, skills, and organisational constraints a justified belief about model correctness.* Given such definition everything hinges on what concept is conveyed by correctness. M&S communities used to keep distance from the initial definition. The closest definition to systems engineering found in the literature is: *The property that a model is correctly represented and was transformed correctly from one representation form into another, according to all transformation and representation rules, requirements, and constraints.* (Brade 2004). The author depicts correctness by consistency – correctly described in all their different representation forms – and completeness – completely consistent with each other.

Validation and validity

In system engineering, validation is the activity that consists to make sure, essentially by tests means, that a product is conforming to its specification or the product is satisfying the user requirements. The M&S tradition associates validation as the activity to make sure that a model or a simulation has behaviour indistinguishable to the system of interest. Validation deals with the examination of fidelity (Pace 1999), suitability (Brade 2004) or validity (METHGU2 2004). In (METHGU2 2004) the definition of validation is *the process which is used to construct, under a set of time, cost, skills, and organisational constraints a justified belief about model validity.* In fact the main problem with validation is to clarify the concept of validity for a specific context. For instance in (Sargent 1987), the validity is depicted in model validity, data validity and conceptual model validity. These concepts are respectively related to model use validity, data validity and model validity in (ITOP 2003). (METHGU2 2004) rightly assimilate model use validity to experimental frame validity.

Decision and quality assurance

Quality assurance is defined as (1) “a planned and systematic pattern of all actions necessary to provide adequate confidence that an item or product conforms to established technical requirements.” (2) “a set of activities designed to evaluate the process by which products are developed or manufactured.”(IEEE 2002)
To provide the adequate confidence in a product, quality used to be decomposed in:

- The product intrinsic quality: ability to satisfy the user needs.
- Quality of the product development: product elaboration of which the quality can be proven while mastering efforts.

Simulation results must only be used, if they are sufficiently credible with respect to the impact of their use, and the influence of the simulation results in comparison to other non-M&S influences (“conventional” information). If the influence of the model, simulation results, or observed model behavior is high, wrong or unsuitable simulation results are not compensated by conventional information, and most probably lead to wrong decisions with undesired consequences.
Also, convince an evaluator/user to apply our recommendations for an M&S product leads to more than two qualities more or less steady:

- We really reach the initial objectives of the study.
- We reach the objectives and it is cheaper than other ways.
- We demonstrate that simulation uncertainty is low enough to avoid identified risks and impact…

These concepts can vary according to particular aspects of M&S project, objectives, risks identified related to the project… Such constraints have led to provide framework that encourages users to clearly identify the properties that the simulation model requires to be considered as valid for their intended purpose and identify the *evidence* that they would like to see to believe that the simulation model is valid. In that sense the deployment of adaptable VV&A processes became a perspective of study. The claim-argument-evidence structure (ITOP 2003) and the REVVA Generic Process (METHGU2 2004) widely contribute to such framework.

FRAMEWORK FOR VV&A M&S

The framework presented here is based on quality assurance items, the claim-argument-evidence structure (ITOP 2003) and the REVVA Generic Process (METHGU2 2004). It aims to provide a logical structure for presenting V&V results and understand their influence. The following framework should not be used as it is. As a starting work, we could not be exhaustive in our considerations to cover all VV&A aspects and subtleties. However it shows that existing standards must be considered to increase VV&A practices.

Quality assurance items

Criteria
Broadly speaking, a criterion is “an established standard by which something may be judged or examined” (New Mexico). VV&A criteria are used for guidance in evaluating the adequacy of a simulation product. The set of criteria that will be selected will aims to increase the objectivity and the consistency of the final decision. Criteria could be refined into sub-criteria. A criterion is user oriented and cannot be directly measured.

Factors
In fact the criteria should be refined until it is assumed that it is directly measurable. The factor, which can be seen as a specialized criterion, is product and process oriented and directly measurable.

Metrics
In software quality, a metric is “a quantitative measure of the degree to which software possesses a given attribute which affects its quality” (IEEE 2002). VV&A measurable criteria (factors) will be associated to metrics in order to provide a quantitative measure for the assessment of a given criteria.

Techniques
Metrics will be measured from different techniques. Numerous techniques can be used for V&V of M&S products including requirements engineering and software engineering. (DMSO 2000) lists and classifies some of these techniques. (FDA) lists also numerous software testing techniques.

Metamodel[2] for VV&A

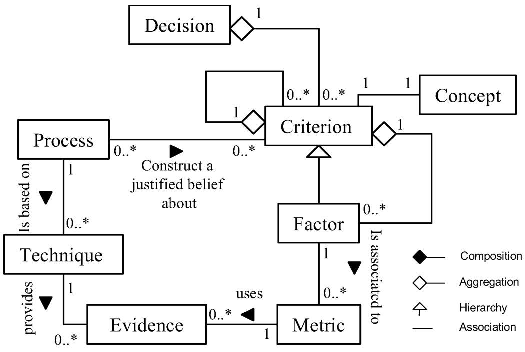


Figure 4. VV&A Metamodel

The metamodel is build with UML class diagram (UML-RM and UML-UG 1999). Briefly:
- Inheritance, refers to the ability of one class (child class) to *inherit* the identical functionalities of another class (super class), and then add new functionalities of its own.
- An association is a link between two classes. The association has a name (e.g. is based on), ▶ is the lecture sense.
- Aggregation is a special type of relationship used to model a "whole to its parts" relationship. In basic aggregation relationships, the lifecycle of a *part* class is independent from the *whole* class's lifecycle.
- The composition relationship is just another form of the aggregation relationship, but the child class's instance lifecycle is dependent on the parent class's instance lifecycle.
- Association has cardinalities. This notion is similar to the one in Merise method. In UML we speak about multiplicity values.

The figure 4 is illustrating by a metamodel the concept of decision with:

- Decision: the decision to say that we accept or not the M&S asset for the intended use.
- Criterion: the criteria that the M&S product needs to meet to be acceptable for its intended purpose (e.g. correctness, utility, fidelity, validity…). Note that a criterion can be an aggregation of its child’s sub-criteria.
- Concept: the precise definition about which idea is conveyed through a criterion.
- Factor: a directly measurable criterion (e.g. FM consistency, FM to EM completeness, model I/O behaviour goodness of fit, code legibility…).
- Metric: a quantify point of measure for a specific factor (e.g. for factor EM consistency, a metric could be the number of bugs per line of codes).
- Techniques: quantitative or qualitative effort which supports process and provides evidence for metrics (inspection, subject-matter-expert opinion, review, testing…).
- Process: Based on several techniques, processes *construct a justified belief about* (METHGU 2004) criteria (e.g. verification, validation, V&V evaluation, model development process evaluation…)
- Evidence: V&V based techniques results which support the criteria assessment (e.g. traceability matrices, coverage rate, technique cost in term of time/money/personnel, techniques or results influence level, techniques or results conviction degree…).

With UML it is possible to assign attributes to classes and instances of these classes. An attribute give information about the object state. Figure 5 illustrates this concept. The attribute section of a class (the middle compartment) lists each of the class's attributes on a separate line. Each attribute line uses the following format:

name : attribute type

For example, a technique may have a lingering degree of conviction range from 0 to 10 (according to the type of technique it is – formal or informal)

influenceDegree : Integer

Further, a specific criterion would have an influence level different from another criterion related to its level of impact, level of V&V or level of quality. These kinds of consideration may be interpreted as Criterion’s attributes and

[2] In computer science and related disciplines, metamodeling is the construction of a collection of "concepts" (things, terms, etc.) within a certain domain. A model is an abstraction of phenomena in the real world, and a metamodel is yet another abstraction, highlighting properties of the model itself. This model is said to conform to its metamodel like a program conforms to the grammar of the programming language in which it is written.

would influence the aggregation to its higher level super-criterion. These attributes are given as example. For instance (LEVELS 2004) defines the level of impact and addresses the identification of levels of V&V rigor and intensity that can be used as appropriate indication of the residual uncertainty associated with the process of V&V. Many others have still to be identified. The objective is to provide identity cards for every element involved in M&S VV&A.

Like attributes, the operations of a class (lowest compartment) are displayed in a list format, with each operation on its own line. Operations are documented using the following notation: name(parameter list) : type of value returned

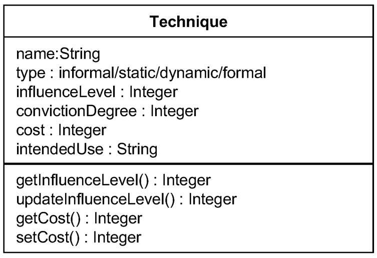

Technique
name:String
type : informal/static/dynamic/formal
influenceLevel : Integer
convictionDegree : Integer
cost : Integer
intendedUse : String
getInfluenceLevel() : Integer
updateInfluenceLevel() : Integer
getCost() : Integer
setCost() : Integer

Figure 5. Class Diagram of a Technique

The benefit of such an approach is to be able to dynamically change attributes and to have an ongoing visibility on them. For example a specific technique which has proven many times to be most efficient than another for a particular problem to solve could see its degree of conviction increased. The difficulty is to be able, when the final decision is taken, to find out which elements within the all M&S VV&A elements was responsible to such an asset.

Metamodel for M&S

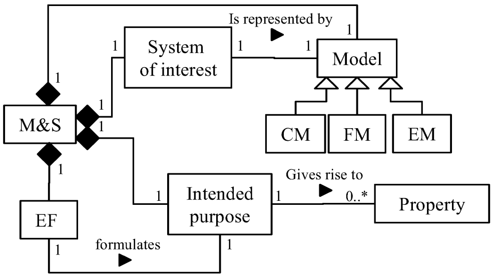


Figure 6. M&S Metamodel

Figure 6 illustrates the concept of an M&S with:

- M&S: the final Modelling and Simulation products.
- System of interest: the real world <u>subjected to modelling and simulation.</u>
- Model: the representation of the system of interest <u>within a specific experimental frame of interest</u>. A model can be of various forms (conceptual model, formal model, executable model or even simulation results).
- Intended purpose: the objectives of simulation use. The constraints and risks identified related to the simulation should also be addressed. The intended purpose is user-oriented.
- EF: the operational formulation of the intended purpose.
- Property: properties that the simulation model requires to be considered as valid for the intended purpose.

Metamodel for M&S VV&A

Figure 7 illustrates the concept of an M&S VV&A.

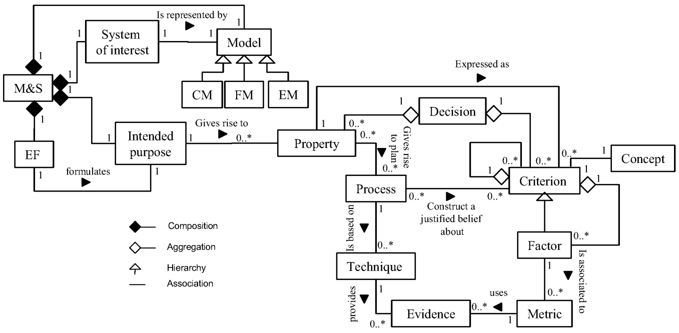


Figure 7. ¨M&S VV&A Metamodel

Properties that the simulation model requires to be considered as valid for the intended purpose are expressed as criteria and give rise to plan processes.

APPLICATION

By "instance" of the M&S VV&A metamodel we can produce a specific VV&A planning. We have used UML collaboration diagram (UML-UG and UML-RM 1999) to realize it (figure 8). Collaboration diagrams are particular cases of interactions diagram which represent a dynamical view of the system. They introduce a set of roles played by objects in a particular context and relationship between these objects. They particularly make an issue on the spatial structure which allows building up collaboration within a set of objects. The temporal dimension requires sequence number definition for messages.
The roles and responsibilities of people involved in M&S VV&A are not so simplistic. More details about this topic are effectively described in (METHGU2 2004).

This example is based on VV&A concepts issued from (LEVELS 2004).where:

- The level of overall residual uncertainty quantifies the uncertainty associated with the perception of correctness and validity of the M&S product, based on the rigor of criteria identification and the intensities of their individual substantiations. The level of residual uncertainty is defined in dependence of the level of V&V rigor and the level of V&V intensity.
- V&V rigor depends on the examined variety of desired model properties and the coverage of the intended purpose during V&V. Rigor addresses the completeness of criteria and the inferability of the decision from the examined V&V objectives.
- V&V intensity depends on the objectivity, thoroughness, exhaustiveness, and repeatability of

the assessment of the examined criteria and V&V objectives.

In a collaboration diagram, objects are interacting through messages which appeal their methods. When an object creates another object, it creates also a relationship between them or with others objects. (e.g. create(rigor, uncertainty) means that rigor is a uncertainty sub-criterion. The method add() assigns a sub-criterion to a super-criterion).

1: User precisely identifies the properties that the simulation model requires to be considered as valid for its intended purpose and to express them as criteria,
create()
create()
2.1: User expresses properties as criteria – create(Criterion)
create(utility)
create(uncertainty)
If required user identifies sub-criteria – create(Criterion,Criterion)
create(model validity, utility)
create(model development quality, utility)
create(rigor, uncertainty)
create(intensity, uncertainty)
2.2: User identifies the evidence that he would like to see to believe that the simulation model is valid – create(Evidence)
create(traceability matrix between real system I/O and model I/O)
create(traceability matrix between properties and criteria)
create(cost)
3: Provider identifies required factors to support criteria evaluation – create (Factor, Criterion)
create(I/O behavioural goodness of fit, model validity)
create(level of effort, model development validity)
create(criteria completeness, rigor)
4: add(Factor)
5: Provider identifies metrics that will serve as a quantify point of measure for factors and that will be supported by evidences – create (Metric, Evidence, Factor)
create(traceability between real system I/O and model I/O, traceability matrix between real system I/O and model I/O, I/O behavioural goodness of fit)
create(traceability between properties and criteria, traceability matrix between properties and criteria, criteria completeness)
create(global M&S cost, inspection cost, review cost, traceability cost, level of effort)
6: associate(Metric)
7: Provider plans the processes required to construct a justified belief about criteria – create (Process, Criterion)
create(validation, model validity)
create(model development evaluation, model validity)
create(requirement analysis, rigor)
8: associate(Process)
9: Provider identifies the required techniques to support processes – create (Technique, Process)
create(inspection, validation)
create(review, model development evaluation)
create(traceability assessment, requirement analysis, traceability matrix between properties and criteria)
10: associate(Technique)
11: Provider conducts technique to provide evidences – set (Technique, Evidence)
create(inspection, traceability matrix between real system I/O and model I/O)
create(review, cost)
create(traceability assessment, traceability matrix between properties and criteria)
12: VV&A supervisor assembles and integrates the approved evidences supporting the metrics – aggregate (Evidences, Metric)
aggregate(review cost, inspection cost, traceability cost, global M&S cost)
aggregate(traceability between real system I/O and model I/O, traceability matrix between real system I/O and model I/O)
aggregate(traceability between properties and criteria, traceability matrix between properties and criteria)
13: VV&A supervisor assembles and integrates the metrics to evaluate factors – aggregate (Metrics, Factor)
aggregate(global M&S cost, level of effort)
aggregate(behavioural goodness of fit, traceability between real system I/O and model I/O)
aggregate(criteria completeness, traceability between properties and criteria)
14: VV&A supervisor assembles and integrates the factors to evaluate criteria by using specific techniques for aggregation such as multi-criteria decision making (MCDM) (Zeleny 1982) until reach the high level criteria – aggregate (Factors, Criterion)
aggregate(global M&S cost, model development quality)
aggregate(behavioural goodness of fit, model validity)
aggregate(criteria completeness, rigor)

aggregate(uncertainty, rigor, intensity)
aggregate(utility, model validity, model development quality)

15. The final decision can be taken between users and VV&A supervisor – aggregate (Criteria)
aggregate(utility, uncertainty, property)

16. If necessary VV&A supervisor can update some elements attributes (level) based on empirical observations and feedbacks on evaluation/decision – update(Level)

As we have seen, the traceability between VV&A items is formalized by the intrinsic notion of relationship provided by a collaboration diagram. Also roles and responsibilities within M&S VV&A are naturally considering.

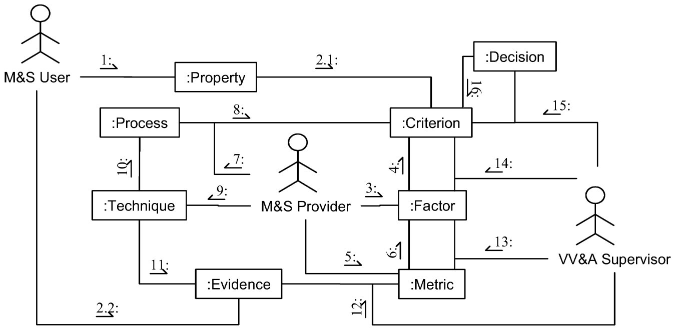


Figure 9. Collaboration Diagram

CONCLUSION

Benefits of such an approach

We have seen that the emergence of standard as UML can contribute to an effective and structured representation of M&S and VV&A concepts. Also such an approach would:
- Increase communication between people involved
- Increase visibility and generalisation for reuse
- Reduce the need for duplicative V&V

- Formalize a structured and comprehensive approach for people interesting in developing VV&A for their M&S enterprise
- Provide a uniform documentation and exchange of V&V information
- Encourage the development of specific languages to specify the related concepts
- Dynamically updating concepts attributes
- Automate V&V Plan, V&V Report
- Manage traceability
- Keep a compact documentation about what has been done and what was the results
- Increase objectivity and so on confident.

Further works

Further work must be accomplished to go deeper into this contribution. As a first every M&S VV&A concepts and their relationship must be well understand and structured. When good foundations will be establish, an issue would be to define domain specific languages to specify a concept. The final issues would be to automate as much as possible. Techniques for aggregation should be identified as well and integrated in the framework. In this contribution we should focus on the VV&A concepts, M&S should be addressed as well. For instance, specific formalisms could be used to describe conceptual model.

REFERENCES

Brade, D. 2004. "A Generalized Process for the Verification and Validation of Models and Simulation Results." Dissertation, Faculty for Computer Science, University of the German Federal Armed Forces Munich, Neubiberg

Defense Modeling and Simulation Office. 2000. "Verification, Validation and Accreditation (VV&A) Recommended Practices Guide (RPG)." Alexandria, VA. http://vva.dmso.mil/

FDA US. Food and Drugs Administration, Glossary of computerized system and software development terminology, www.fda.gov/ora/Inspect_ref/igs/gloss.html

IEEE Standard Glossary of Software Engineering Terminology, Std. 610.12-1990.

IEEE Std 730-2002, IEEE Standard for Software Quality Assurance Plans

INCOSE International Council on System Engineering, "Software Engineering Handbook", INCOSE-TP-2003-016-02, Version 2a, 1 June 2004

ITOP International Test Operations Procedure on V&V Working Group of Experts. 2003 (to appear). International Test Operations Procedure on V&V. Draft, ITSEC, MC7, WG7.2.

LEVELS. 2004: VV&A Levels Definition. THALES report JP1120-WE1300-D1301-LEVELS-V1.0

METHGU2. 2004: VV&A methodological guidelines Reference Manual. THALES report JP1120-WE5100-D5103-METHGU2-V1.0

New Mexico public education department http://www.nmlites.org/standards/language/glossary.html

Pace, D "Development and Documentation of a Simulation Conceptual Model", Fall SISO Interoperability Workshop, Sep. 1999

Sargent R.G., "An Overview of Verification and Validation of Simulation Models." Winter Simulation Conference, Atlanta, GA. Eds. A. Thesen, H. Grant and W. D. Kelton. Piscataway, NJ: Institute of Electrical and Electronics Engineers, 1987. 33-9.

TOAGUID. 2004: TOA Guidelines. THALES report JP1120-WE2200-D2201-TOAGUID-V1.0

UML-RM, "The Unified Modeling Language Reference Manual." Rumbaugh, J., Jacobson, I. and Booch, G., 1999, Addison-Wesley

UML-UG, "The Unified Modeling Language User Guide". Booch, G., Rumbaugh, J.and Jacobson, I., 1999, Addison-Wesley

Zeigler, B.P., H. Praehofer, and T.G. Kim. 2000."Theory of Modeling and Simulation". Second Edition. Academic Press. ISBN: 0-12-778455-1

Zeleny, M. Multiple Criteria Decision Making. New York: McGraw-Hill, 1982.

MODEL INTEROPERABILITY

PROCESS-INTERACTION DIAGRAMS FOR STRUCTURED DISCRETE-EVENT SIMULATION MODELING

Acácio M. O. Porta Nova
Department of Engineering and Management
Instituto Superior Técnico
Av. Rovisco Pais
1049-001 Lisboa
Portugal
E-mail: apnova@ist.utl.pt

KEYWORDS
Simulation, discrete-event, modeling, process-interaction, diagrams.

ABSTRACT

In this paper, process-interaction diagrams are proposed, for both teaching and practicing structured discrete-event simulation modeling. This approach is compared to existing graphical description tools for building simulation models. The main aspects are illustrated using simple queueing systems.

INTRODUCTION

Being involved with simulation for almost 30 years, I have been actively witnessing the evolution in languages and methodologies for discrete-event simulation modeling (Nance 1996). However, developments driven by graduate research and archival publication have raised formal model specification to a level of abstraction and sophistication that is virtually unreachable, except to computer science experts. Although no one can argue the importance of such an approach in very large-scale software engineering projects, like the production of yet another simulation language, or visual development environment, we can certainly question its relevance for many simulation projects at the company level. On the other hand, in spite of the intimate relationships between simulation and computers, since their common inception in the mid 40s, many useful concepts and methodologies from structured programming (Dahl, Dijkstra and Hoare 1972) do not seem to have pervaded simulation modeling. In fact, top-down development, manageable modules (or blocks), stepwise refinement, simple control structures, model understandability, documentation, all these concepts seem as up-to-date in the simulation context today, as they were for general purpose programming three decades ago. It is never excessive to emphasize the importance that a structured modeling approach can have for a newcomer to the field of simulation. However, even a simulation expert can profit from the use of such an approach in the initial steps of building an eventually complex simulation model. Intensive use of graphical description languages and postponing as much as possible the coding of actual commands were also keystones to sound programming style. This is especially appropriate in a field that gave birth to the object-oriented paradigm, with the language Simula 67 (Nygaard and Dahl 1981). Indeed, no architect starts an ambitious project with a detailed, rigorous (technical) drawing of a building, bridge or monument. Rather, their initial sketches might even be (and sometimes they are) casually discarded as childish, or child-produced. After all, we are quite familiar with that old Confucian saying that a picture is worth a thousand words…

Let us consider now the teaching of simulation modeling at the university level. A rather disturbing development, inescapable to anyone involved with teaching science and engineering curricula is the steadily decreasing number of student candidates. Many of them also lack the background or motivation to go much further than the chase for easy problem-solving recipes. The ever-increasing gap between theory and practice makes it harder to fit a meaningful simulation curriculum into an undergraduate course that will often have to cover other operations research topics. Now with the Bologna Process being implemented in universities all over Europe, it is even more important to focus on the core conceptual topics and welcome any measure aimed at increasing the motivational levels of our students. Based on my own experience, I feel that simulation modeling should be discussed as early as possible (I do that immediately after the introductory material). I also think that all three modeling perspectives should be analyzed and practiced for some simple exercises. In fact, we can never be sure of the simulation (or general purpose) programming language that we will have to use next. The use of graphical description tools should be emphasized, before doing any laboratory work with computers and actual simulation languages.

In the next section, two types of diagrams used for simulation modeling are reviewed. Then, process interaction diagrams are proposed for describing process-oriented models. Finally, some conclusions are drawn about this work.

DISCRETE-EVENT MODELING

Within the simulation community, for long there has been a well-established consensus about three *world views*, or *modeling perspectives*: *event-oriented*, *activity-oriented* and

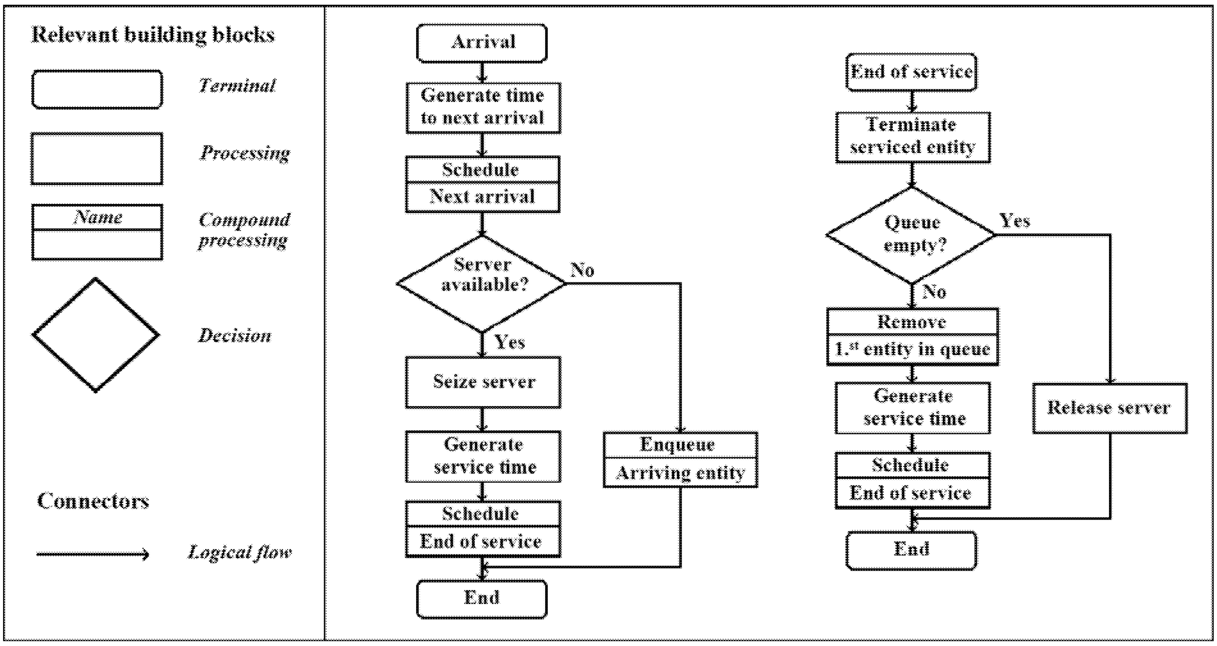


Figure 1: Flowcharts for the Relevant Events of an Event-Oriented Model of the M/M/1 Queue

process-oriented (Kiviat 1967, 1969). These perspectives provided the conceptual framework for the simulation programming languages that were developed since the advent of computers. Thus, a certain simulation model could be equivalently constructed using, as building blocks, either *events*, or *activities*, or *processes*.

Closer to procedural programming, event-oriented languages appeared first and the classical (and now ubiquitous) *flowchart* was the users' choice for developing and documenting event-oriented models. Trying to be parsimonious, I recommend the usage of a selected subset of the available flowchart symbols. Two options deserve some justification. The distinction between "processing" and "compound processing" is somewhat arbitrary and it depends on the languages under consideration. Anyway, the first block is usually implementable with a single (high-level) language command, while the second block requires several commands. In general, "compound processing" is either an intrinsic, or a user-defined "name"d procedure. On the other hand, I am strongly against the use of a specific symbol for making "comment"s, or remarks. These graphical model description tools are especially useful for roughly sketching a model and only free-format remarking can fulfill our needs for model description. The flowchart symbols that I recommend are shown in Figure 1, as well as the relevant events (*arrival* and *end of service*) for a simple model of an M/M/1 queue. We omit, from this discussion, implied details, like the language executive, event calendar, starting conditions, or simulation duration.

Less efficient but easier to program, activity-oriented languages have made frequent use of *activity* (or entity) *cycle diagrams* (ACD). An early reference is (Hills 1971). This is a tool that has the smallest possible alphabet of building blocks (two): a rectangle for *active states* (or activities) and a circle for *passive states* (or queues). We recall the symbols used for drawing ACD in Figure 2. Simple rules also try to make it easier for the user to create, or debug, an ACD. Active and passive states should *alternate* (if needed, with the inclusion of *dummy*, or *notational* queues). Also, the only condition that is required for initiating an active state is the availability of an entity in each of the queues immediately preceding that active state. An ACD for an activity-oriented model of an M/M/1 queue is also represented in Figure 2. One of the main drawbacks of ACD is the modeling of "arrivals", as can be observed in Figure 2. Since there is no creation or destruction of entities, all entities to be processed in any simulation run must first reside in an "outside" queue. In order to prevent the simultaneous arrival of all these entities, an artificial entity, a "door", has to be introduced. Focusing on the activities that can occur in a system, ACD naturally lead to activity-oriented models. However, due to its simplicity, the AC Diagram can be very useful in the initial drafting of models, regardless of the modeling perspective implemented in the simulation language to be used. For years, I have been emphasizing the use of these diagrams in the undergraduate simulation courses I have been lecturing, although we actually use a process-oriented simulation language. Nevertheless, ACD do have their idiosyncrasies:

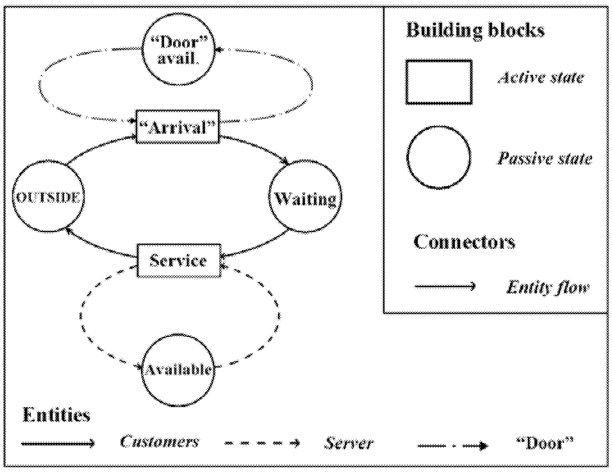


Figure 2: Activity Cycle Diagram for an M/M/1 Queue

notational queues, arrivals modeled as activities, no creation or destruction of entities, and so on.

With respect to process-oriented languages, although they were the last of the three families to appear, they are the *de facto* standard nowadays. In general, they use a visual programming approach, based on graphical user interfaces and symbolic commands. Process-oriented models implement the more intuitive approach of the naïve modeler, describing the life cycle of each relevant entity. In this case, no simple commonly accepted graphical description tool emerged, although many different types of diagrams have appeared in the literature. Of course, we can argue that the languages themselves are graphical, and that we can use the specific (graphical) symbols or commands of the selected language itself. However, this is akin of defending that the best way to build an algorithm is using directly the commands of the actual programming language, a discussion that has been rendered obsolete by the emergence of structured programming a few decades ago. That does not seem to be the most adequate conceptual approach and it makes us too dependent on the specific software and its supplier.

PROCESS-INTERACTION DIAGRAMS

I think that anyone interested in using simulation will feel very comfortable describing event-oriented models with classical flowcharts. In fact, they are even used in some contexts for which they are not appropriate at all (for example, within the process-oriented framework). In many small corporations, when specialized (and expensive) simulation software is not available, young graduates still use flowcharts to prepare the programming (from scratch), in a general purpose language like C, of their own event-oriented models. On the other hand, when a (any) simulation language is available, before we decide whether our entities are going to be modeled as resources, facilities, transactions, or whatever, it is a good idea to identify first which ones are actually relevant for the problem we have to solve. I never found anything more useful that an ACD for this purpose: with only two symbols available, we are forced to focus on the entities, activities and queues that are relevant for our problem. Of course, later on, when we try to add realistic behavior to the ACD, the limitations of the tool start to hinder our efforts. Thus, I always felt that we needed a graphical tool for this phase of the development process of a simulation model. It should be adequate for the process-oriented world view, but compatible with the flowchart and the ACD. In syntony with the structured programming guidelines, I tried to use simple control structures and to avoid re-inventing the wheel. These constraints lead to what is proposed in this paper. In this humble contribution, a significant role can be traced back to my lecturing of industrial engineering and operations management courses.

The proposed building blocks and connectors for constructing process-interaction diagrams (PID) are represented in Figure 3. The *source* is naturally used to create entities. As its name implies, the *queue* represents a conceptual location where an entity has to wait, before being able to advance in its process (the *notational* queues used in ACD are to be avoided here). A *process component* is any relevant part (more or less detailed) of an entity's process (at the lowest level, it can represent an activity or an event). The *drain* (or *sink*) terminates the entities, when they are no longer necessary in our model. Except for the process component, all other blocks are used to describe the process of a single type of entity. In a process component, several types of entity may interact, originating different types of transformations. Consequently, entities may be created, modified, assembled, separated or terminated. Two types of connectors may link the blocks in a PID: a (unidirectional) *thin line* represents the flow of entities in their own processes; a (uni-, or bidirectional) *thick line* represents the *influence* of a *process component* on another *process component* (eventually in different processes).

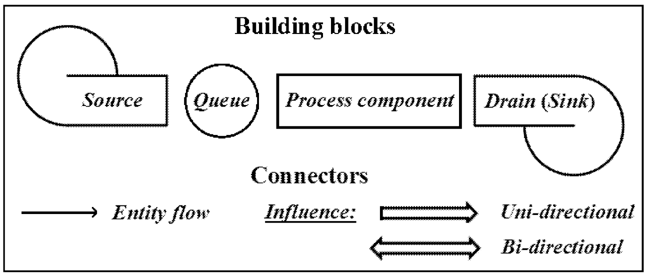


Figure 3: Symbols for Process-Interaction Diagrams

Using the above symbols, it becomes very easy to build a PID for an M/M/1 queue that we have been using to illustrate our discussion; see Figure 4. However, in contrast to ACD, PID allow hierarchical modeling. Suppose that Figure 4 is actually a high-level representation of (loaded) crude oil tankers that are contracted in the spot market to unload at a nearby port supplying a refinery. Our target system would include the processes of the tankers and the processes of the port resources used by the tankers. Under these conditions, "service" would represent the period of time required by the tankers to complete the unloading operations, after the assignment of a docking berth. Thus, the "server" would be the berth reserved for the docking of the tankers. We could then detail the "service" *process component* (the dashed rectangle in Figure 4) as shown in Figure 5 (again, the dashed rectangle). In addition to detailing the processes of the two previously defined entities, a new one (a tug) was included. This was done because docking or undocking a tanker required the assistance of a dedicated tug. The inspiration for this example came from a port in northern Portugal (Leixões) that also had another troublesome characteristic: some storms might provoke the interruption of the unloading operation. When this happened, the tanker had to be

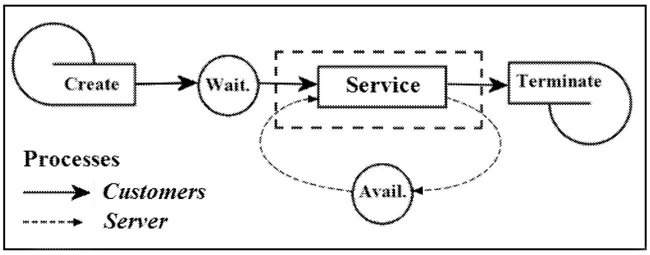


Figure 4: Process-Interaction Diagram for an M/M/1 Queue

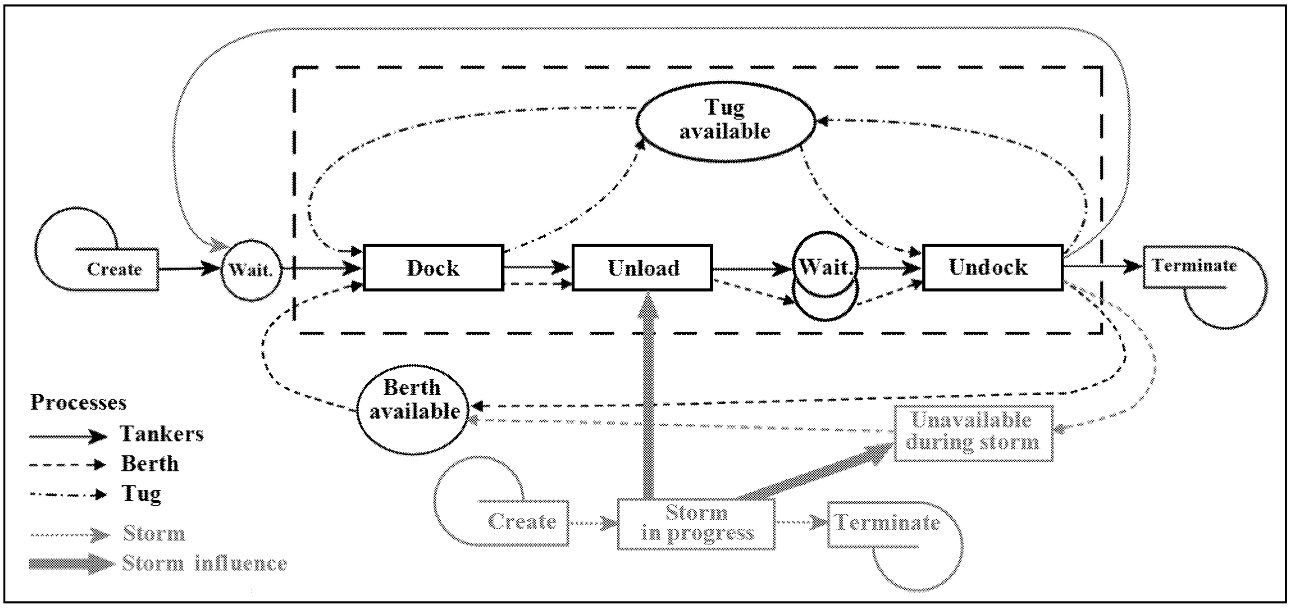


Figure 5: Complete Process-Interaction Diagram for the Port Operations

undocked and wait in open sea until the end of the storm. It would then dock again, to resume unloading. Even this reasonably complex modification can be inserted into the PID without changing any of the elements that had already been included in the previous draft of the model; see Figure 5 for a complete PID of these port operations. Another entity (the storm) was added to our model, with its own (separate) process. The difference now is that when a storm is in progress it *influences* both the unloading operation and the period during which the berth is unavailable for docking. Some simulation languages use a *preempt* command for this purpose. We represent the required modifications in red color, in order to make them more visible.

Another somewhat controversial modeling issue is the hypothetical inclusion of materials (to be processed) in the model description diagrams (e.g., in ACD). Sometimes, this decision seems to be based more on prestidigitation arts than easily "student-comprehensible" objective reasons. In my opinion, if they need to be explicitly manipulated (as entities, or even as global variables) during the simulation, they should be included in the model diagram. For instance, in the port operations described in Figure 5, *crude* is only implicitly modeled. Thus, we are assuming that it suffices to model the unloading operation as a simple duration (obviously proportional to the amount of crude transported by the tankers).

We all know too well that many simulation models, at a certain point, require logic so intricate that many simulation languages provide links to procedures written in general purpose programming languages, like C, or Visual BASIC. Even these combined models are compatible with PID: the corresponding *process component* (in the PID) would represent an event, whose logic could be described using a classical flowchart, of the type represented in Figure 1.

CONCLUSIONS

In this paper, we propose a new graphical description tool for process-oriented simulation models that significantly extends existing approaches. Process-interaction diagrams allow for hierarchical modeling and are able to expose substantial portions of the model logic. This approach was illustrated using a real-life situation. Actual coding of the diagrams should also be straightforward in most existing process-oriented languages.

REFERENCES

Hills, P.R. 1971. *HOCUS.* P. E. Group, Egham, Surrey, England, U. K.

Kiviat, P.J. 1967. "Digital Computer Simulation: Modeling Concepts". RAND Memo RM-5378-PR. RAND Corporation, California.

Kiviat, P.J. 1969. "Digital Computer Simulation: Computer Programming Languages". RAND Memo RM-5883-PR. RAND Corporation, California.

Nance, R.E. 1996. "A History of Discrete Event Simulation Programming Languages". In *History of Programming Languages, Volume 2*, T.J. Bergin, R.G. Gibson, R.G., Jr Gibson (Eds.). ACM Press, New York, 369-427.

Nygaard, K., and O.J. Dahl. 1981. "The Development of SIMULA Languages". In *History of Programming Languages,* R.L.Wexelblatt (Ed.). Academic Press, New York, 439-491.

AUTHOR BIOGRAPHY

ACÁCIO M. O. PORTA NOVA is an Associate Professor in the Department of Engineering and Management at the College of Engineering (Instituto Superior Técnico) of the Technical University of Lisbon. He received a B.S. degree in electrical engineering from IST in 1978. He received a Ph.D. in operations research from the University of Texas at Austin in 1985.

CONTEXTUAL TESTING OF INTERACTIVE PRODUCT SIMULATIONS FOR NEW GENERATION PRODUCTS

Alex Woolley
Steve Gill
PAIPR research group,
National Centre for Product Design and Development Research, ,
University of Wales institute, Cardiff,
CF5 2YB Wales,
Email: awoolley@uwic.ac.uk

KEYWORDS

Information Appliance, Contextual testing, Industrial design

ABSTRACT

Information appliances such as mobile phones, mp3 players and digital set top boxes are becoming ever more pervasive as the available computing power for devices increases. For information appliances that are breaking new ground and entering a market that is new or where the customer is little understood, it is important that customer requirements are gathered to help define the product requirements. Contextual testing using prototypes forms an important part of this process. However, information appliance design presents problems for generating interactive product simulations that are representative of the final device at the early stages of development. An interdependence of bespoke hardware and software is restrictive of iterative development as it is difficult to bring both aspects together until late in the design process. This can restrict product testing to late on in the design process where large investments of time, personal and financial capital have already been made.

This paper presents a literature review of the challenges and available solutions to generating and testing interactive product simulations of new generation products in the early stages of the design process. It also proposes avenues for further research in this area of design.

INTRODUCTION

New generation interactive products present particular challenges for simulating interactivity as part of the requirements capture process of new product development. The exploratory nature of the concept generation phase of the design process demands flexibility, particularly for new products where the domain is not fully understood. Some of the characteristics of a new generation product identified by Smith (1998) present an insight into why this activity is so challenging:

- *'No clear understanding of user requirements*
- *Involves new or not-yet-existing hardware and software technology*
- *Constantly evolving product features*
- *No comparable existing product to benchmark against'.*

Interactive product simulations (The term "interactive product simulation" will be used to denote a prototype that combines a digital interface with a physical model) for use in user tests forms an important part of the process of ascertaining what an information appliance should be and is crucial to solving the problems highlighted by Smith. Schrage (2000) quotes David Kelly of IDEO as saying innovation cultures need to 'move from spec-driven prototypes to prototype-driven specs', adding weight to the argument that testing of rapid prototypes in context should form a part of the early, exploratory requirement capture phase of product development.

Driving development by testing prototypes is a central theme of human centred product development, i.e. studying users to generate product requirements and gathering feedback from them throughout the design process. Holtzblatt (2005) describes the customer-centred approach as contextual design. This approach to designing puts the user at the heart of the process and encourages constantly gathering user data to both inform the initial design requirements and to iterate towards a solution.

The problem for achieving this as part of information appliance design is that information appliance design is at the crossroads of a series of design disciplines. Designers 'are no longer bound by the generic technology offerings of a PC' (Mohageg and Wagner, 2000) and there is immense freedom available to the designer in the way product interfaces are controlled and displayed. Input is needed from Industrial Designers, GUI Designers, Software Engineers, Mechanical Engineers and Electronic Engineers – making information appliance design a multidisciplinary process. Norman (1998) highlights that it 'often takes three people to cover the capabilities required' to rapidly prototype an information appliance. There are a number of prototyping solutions available to solve this problem by enabling a more rapid generation of interactive simulations. These will be reviewed later in the paper in relation to contextual data gathering.

THE IMPORTANCE OF CONTEXTUAL DATA GATHERING

Traditional market research data has focused on identifying demographics and providing a statistical evaluation of the marketplace. Although this data is useful, product design has seen a rise in other techniques that provide a richer picture of the consumer to provide a more in depth view to support statistical market research data. This is particularly true for the consumer goods sector where product differentiation in the marketplace is important. Product Design has recently seen a rise in the use of ethnographies to generate a picture of the consumer (Norman 1998). This is often turned into a literal picture forming a persona (Don and Petrick, 2003) to create a personality with a set of values that typifies the target consumer. Sacher and Loudon (2002) emphasise the use of ethnographic methods as part of culture-based design where development starts by uncovering shared values and beliefs of a culture group to provide guidelines for design.

Rob van Veggel's (2005) description of the history of the shift in design towards ethnography provides some interesting insights into the implications for moving prototype testing away from the usability laboratory and into the real context of use:

'Initially, (designers) turned to psychology, which has limited application. First of all, psychologists develop their understanding by performing tests in controlled environments such as labs. The resulting knowledge often is too general, too abstract, and too much divorced from real life situations, and therefore difficult to apply in actual situations targeting specific customers. Second, psychologists primarily approach humans as individuals.'

The drivers behind the move to ethnography in design are important to consider in respect to the user testing of information appliance prototypes. Laboratory-based usability testing of information appliances, many of which are inherently mobile, provides data that is 'divorced from real life situations' and similarly treats users and indeed the information appliances 'as individuals'. Talking of information appliances as individuals may seem like an overextension of the metaphor, but the Media Equation proposed by Reeves and Nass (1998) suggests that this is not the case. Reeves and Byron present the case in compelling depth that 'Media = Real Life', that we treat computers as real people, and that the same social rules apply. Therefore isolating an information appliance prototype in a controlled environment and attempting to use it to generate product specifications is liable to produce generalised, unspecific data and not the rich insights needed for David Kelly's proposed prototype driven innovation. This is reinforced by Buchenau and Suri's (2000) view that 'the experience of even simple artefacts does not exist in a vacuum but, rather, in dynamic relationship with other people, places and objects'.

USER TESTING WITH INFORMATION APPLIANCE SIMULATIONS

Laboratory based interactive product simulation testing

There are various quantitative methods appropriate for the statistical analysis of interface simulations. Techniques such as studying optimal paths, error analysis, and verbal protocol analysis (Nemeth, 2004) provide excellent data for identifying bugs or errors in interface architecture and ensuring that the design team's mental models match those of the users. Quantitative methods tend to be suited to laboratory based testing where variables can be controlled; whether in a fully blown usability laboratory as described in Rubin (1994) or simply in a room appropriated for a day's testing. Although laboratory testing is often 'caricatured' in literature depicting 'user testing as videotaped laboratory tests with set tasks and no context' (Hertzum, 1999) it is fair to say that the general ethos of controlling variables to provide robust data for statistical analysis is a general principle of laboratory based usability testing. This can present problems for information appliance testing, particularly as many of the devices are inherently mobile, as they are intended to be used in an uncontrolled and changing environment. This elimination of environmental and emotional factors can affect how results correlate to the reality of operating the product. Norman (2004) describes how users who are stressed are less likely to solve problems creatively and are likely to approach them in a linear fashion. Ware (2004) discusses how under stress the user's useful field of vision becomes restricted, limiting the awareness of other events. Factors such as this have important implications for interface design, particularly in the case of devices where correct performance is critical and or the environment is extreme.

Many laboratory tests are also scripted with the user undertaking a prescribed set of activities. This allows the generation of larger volumes of statistically valid data and also allows the prototype to have only certain sections of the interface functioning. The problem with tight scripting is that it prevents the exploration of whether the scripted task is even the correct task to be undertaking.

Integrating interactive simulations in contextual testing

It is possible to integrate interactive product simulations into some of the more ethnographic approaches to user testing. Interviews (Holtzblatt, 2005), storyboarding techniques and Scenarios (Carroll, 2000) can all be adapted to include an interactive prototype. Again, it is likely that there will need to be an element of scripting of tasks with this kind of study in a similar way to laboratory testing due to limits on the functionality of an early stage prototype. However, conducting the studies in the correct context allows discussion around tasks; helping to discover if in the device is handeling the correct tasks in the correct way, rather than limiting activities to discovering if a user took the optimal path through an interface.

Diary studies (Rieman, 1993) (Palen and Salzman, 2002) have also been used with interactive product simulations to explore how a concept performs in its correct context. This method allows a longer term exploration of a concept than laboratory testing or interviews which may last only an hour or so. The study of an interactive photo frame conducted by van Vugt and Markopoulos (2003) demonstrates how this method can be applied to reasonably low fidelity product simulations to gain an understanding of how a product is used in its correct context.

At the earlier stages of design, the exploratory nature of the qualitative methodologies lends them to gaining rich insights into the customer and how a product will fit into their lives. This approach is more appropriate than the tighter scripting of the qualitative methods for gathering user requirements.

A REVIEW OF EXISTING APPROACHES TO GENERATING INTERACTIVE PRODUCT SIMULATIONS

There are a range of solutions available for generating a product simulation combining a physical prototype with a digital interface prototype. Different prototyping tools lend themselves to certain testing methods. For example; prototyping tools intended for laboratory based error analysis testing may well not allow the study of subtle contextual issues such as those investigated in Murtagh's (2002) study of body language and eye contact during mobile phone use in train carriages.

As has been discussed, the multidisciplinary nature of information appliance design makes it difficult to explore product solutions with fully interactive prototypes at the early stages of product development. So why simulate the product in its entirety at all? Although Sharp (1998) showed that virtual simulation on a touch screen was highly effective in predicting interaction performance for his studies of microwave ovens; the PAIPR group (Evans and Gill, 2006) has recently collected data that suggests that the same is not true for hand-held products. By combining both the digital and physical interface of the prototype a more realistic representation is created.

Editable prototypes

It is important at the early stages of design that prototypes and simulations are low investment, both in terms of time, personal and financial capital to ensure that iteration and exploration of ideas is encouraged. Schrage (2000) suggests that even adding colour to a design is too much quoting Michael Barry of Point Forward as saying 'the minute you lay in colour … you finalise it … You send a cue that it's finished'. There are a range of low fidelity, editable prototyping techniques that are appropriate for generating interactive product simulations.

In terms of digital prototyping, state transition diagrams, paper prototyping and high level coding programs such as DENIM (Lin et al. 2000), HyperCard, Macromedia Director and Flash are often used to generate initial interface prototypes before development moves onto higher quality code in programs such as C++. It is important to ensure that tools used to prototype the interface are as flexible as those used to prototype the physical aspects of the product or rapid development of simulations will be impeded.

Arguably, paper prototyping is the earliest stage technique that combines the digital interface of a product with the physical inputs. It is a very quick and basic technique that is very easily editable. The interactivity is achieved by an interviewer switching paper screenshots over on a physical model when a user has made an input. However, these prototypes require a high level of input from the interviewer who must take the role of the computer and is responsible for emulating functionality. This technique is well suited to understanding very high level interface requirements. However, Liu and Khooshabeh (2003) found that even though paper prototyping is very flexible at the early stages, the level of support needed to use them made them 'insufficient for formal user studies' when compared to an automated digital interface.

Beyond paper prototyping, there are editable solutions that combine both the digital and physical elements of an interface. Pin and Play (Villar et al. 2005) and Switcheroos (Avrahami and Hudson, 2002) allow physical inputs to be arranged simply by pushing buttons into a model or in the case of Pin and Play, a flat substrate. This physical interface can then be connected to a PC running the digital interface on either a screen or, for Pin and Play, projected onto the substrate.

Editable prototyping provides a flexible design tool, and these methods are well suited to some forms of user study. Although editability is desirable in a design tool, it can lead to problems in some forms of user studies. This is particularly true if a product is to be tested in its real environment, where having switches simply attached by pins may well result in prototypes that are not robust enough to survive the testing.

A more permanent test object

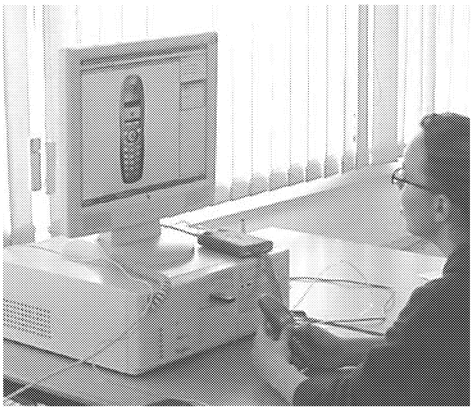


Figure 1: Example of an Interactive Product Simulation made using the IE System under test in Laboratory Conditions

The IE unit (Gill, 2003) and Buck device (Pering, 2002) offer more permanent solutions with switches embedded in a higher quality physical model and then connecting via a cable to a PC. The Buck is restricted to retrofitting old hardware, whereas the IE unit provides an alternative solution enabling switches to be embedded in high or low fidelity models (Woolley and Gill, 2006). One of the potential difficulties with the IE and Buck systems is the lack of an integrated screen (Figure 1). PAIPR has recently gathered data that a model running an interface on a laptop screen provides similar usability data (Evans and Gill, 2006), using the approach recommended by Molich (2002), to a real product. However, when studying some of the more contextual product issues such as how the device is used as part of interaction with other people, this may become problematic. Nam and Lee (2003) has developed a system along similar lines to the IE unit that includes support for projecting the display onto models. Although this is a very flexible way of solving the problem in terms of supporting many different screen sizes, there are limitations in taking the system outside of a usability laboratory; having to remain within a certain proximity to a projector, for example.

The generation of a more robust simulation, particularly of the physical properties of a product allows for wider testing of other aspects of the product. It allows issues such as storage of portable appliances, how products are taken in and out of pockets and bags and how that impacts on the overall experience of the product can all be explored. For new generation products this is an attractive quality as so little is known about users and their requirements.

SIMULATING FUNCTIONALITY

Ultimately, all information appliances have a function. Whether that is to solve a task, communicate or simply entertain. The level of functionality incorporated into a simulation has implications on which methodologies can be used to study the prototype. Incorporating maximum functionality may well allow for broad exploratory data gathering; however this is likely to dramatically increase development times and costs. There are a number of prototyping solutions that allow the integration of increased functionality in terms of sensors and controls available, and also techniques for simulating functionality that can extend level of functionality and therefore the scope of the user testing.

Smarter prototypes

Tangible user interface solutions such as Phygets (2001), MetaCrickets (2000) and the Calder toolkit (2004) allow experimentation with a large range of different inputs and sensors. MetaCrickets in particular allows for a large range of digital and analogue inputs and also some more advanced input sensors. This provides the designer with more choice and the opportunity to generate 'smarter' prototypes. The trade-off for this added complexity is that the designer needs to have more programming knowledge to implement the system and components tend to be physically larger due to more computational power being integrated into the input devices. This added load of size and coding may well prove to be necessary for generating simulations of very complex products that can be tested in the context of use. However, as information appliance design is multidisciplinary in nature, it is important that prototyping tools are accessible to a broad skill set: including those members of the team who are not software engineers. Raskin (2000) states that:

'Programming language environments contain some of the worst human interfaces in the industry…the initial hurdle in terms of system and development environment has become so large that the beginning programmer is not encouraged to learn by doing.'

This is what makes the IE and Buck systems potentially attractive in terms of making tools for multidisciplinary teams, in that the level of coding knowledge needed in order to integrate them with software is extremely low. Both systems work by simulating key press inputs and this enables them to interface with a large range of software, such as Macromedia Flash and PowerPoint with only very simple coding. This shortens the learning curve for designers to begin engaging with interaction issues. This comparatively simple implementation can produce some surprisingly complex and powerful prototypes. However, there is a cost to using key presses as inputs: analogue components such as dials and sliders are difficult to integrate and need to be simulated. In some cases this restricts the performance of these systems.

Wizard of Oz prototyping

The reality of simulating new generation information appliances as Smith discussed is that the concept may involve 'new or not-yet-existing hardware and software technology'. This is particularly true if the information appliance is an interface for a sensor, for example a thermal array in an imaging device, which may need substantial development, and investment, before a sensor can be made for the real product, let alone interface with a smart prototyping kit such as the Calder Toolkit. Wizard of Oz prototyping (Dow et al., 2005) allows functionality to be simulated by the user interacting with what seems to be a fully functional device, but is actually a human being triggering functions from a remote computer. The additional human interaction with the user is less than with paper prototyping and is largely hidden, therefore less likely to influence data. There are also different levels of Wizard of Oz prototyping, for instance most functionality can be performed by the interface but a single aspect of the device such as messaging on a phone can be simulated using Wizard of Oz. For simpler systems such as the IE prototyping method, this presents a viable alternative to integrating high levels of functionality without large investment or complexity at the early stages of the design process.

CONCLUSIONS

The context that an information appliance is used in has important implications for the design of the product, both in terms of performance on tasks when under stress or distracted, and to help define what tasks the product should perform and how. It is important to test in the correct context to help to capture this data, but this testing must be done at an early enough stage to influence the design of the device. This is particularly true for new generation products where so many aspects of the product and the market are not known or not fully understood.

There are a number of existing approaches to generating interactive product simulations and also for testing them in their correct context. However the particular challenges presented by mobile information appliances have not been fully explored or answered and development is needed on both counts of testing and generation of interactive product simulations if a human centred process, where product specifications are driven by prototype testing, is to be fully realised.

FUTURE WORK

This paper has raised several questions for testing information appliances in the context of use. Particularly what tools and methodologies should be used and how much functionality should there be? To start answering some of these questions, a case study is currently being undertaken of a large multinational mobile phone manufacturer to better understand how testing fits into the design process in industry. An additional case study of a design consultancy will also be conducted to compare the process across two different organisational structures. An Action Research approach will then be taken to develop and evaluate a methodology for testing prototypes in the context of use.

ACKNOWLEDGEMENTS

I would like to acknowledge the work in this area of all the other members of the PAIPR group at UWIC and thank them for s this research and for their continuing support.

REFERENCES

Avrahami, D., Hudson, S. E. (2002) Forming Interactivity: A tool for rapid prototyping of physical interactive products; *In: Proceedings of the conference on Designing interactive systems: processes, practices, methods, and techniques*, London, England *June 25 – 28*

Buchenau, M. and Suri, J.F. (2000) Experience Prototyping. *In*: *Proceedings of the conference on Designing interactive systems: processes, practices, methods, and techniques*, 424 – 433, New York City, New York, USA.

Caroll, J. (2000). Five reasons for scenario-based design. *In: Interacting with computers, Vol 13, number 1, 43-60*

Don, A. and Petrick, J. (2003) User Requirements; By Any Means Necessary. *In: Laurel, B. (Ed.) Design Research; Methods and Perspectives*, (pp.70 – 80). London: Cambridge, Massachusetts; The MIT Press

Dow, S. Mackintyre, B. Lee, J. Oezbek, C. Bolter, J.D. Gandy, M. (2005) Wizard of Oz support throughout an iterative design process. *In: Pervasive Computing, IEEE, Vol 4, Issue 4,Oct – Dec, p18-26*

Evans, M. and Gill, S. (2006) Rapid Development of Information Appliances. *In proc of International Design Conference –Design 2006 – Croatia, May 15 - 18*

Gill, S. (2003) Developing Information Appliance Design Tools for Designers. *In: Proceedings of the 1st Appliance Design Conference.* Bristol, UK

Greenberg, S. and Fitchett, C. (2001) Phidgets: easy development of physical interfaces through physical widgets. *In: Proceedings of the 14th annual ACM symposium on User interface software and technology* ; Orlando, Florida *November 11 - 14*

Hertzum, M. (1999). User Testing in Industry: A case study of Laboratory, Workshop, and Field Tests. *Proc. ERCIM Workshop on User Interfaces for All, p59 - 72*

Holtzblatt, K. (2005). Rapid Contextual Design: A How to Guide to Key Techniques for User Centred Design. San Francisco; Morgan Kauffmann

Lee, J. C., Avrahami, D., Hudson, S. E., Forlizzi, J., Dietz, P. H., Leigh, D. (2004) The Calder toolkit: wired and wireless components for rapidly prototyping interactive devices. *In: Proceedings of the 2004 conference on Designing interactive systems: processes, practices, methods, and techniques*, Cambridge, MA, USA *August 01 - 04*

Lin, J., Newman, M., Hong, J., and Landay, J.A. (2000) DENIM: Finding a tighter fit between tools and practice for Web site design. *In*: *Proceedings of the Conference on Human Factors and Computing Systems*, The Hague, Netherlands

Liu, L. and Kooshabeh, P. (2003). Paper or interactive?: a study of prototyping techniques for ubiquitous computing environments. *CHI'03 extended abstracts on Human factors in computing systems*, April 05-10, Ft. Lauderdale, Florida, USA

Martin, F., Mikhak, B. and Silverman, B. (2000) MetaCricket: A designer's kit for making computational devices. *In: IBM systems Journal*, Vol 39, Nos 3&4

Mohageg, M.F. and Wagner, A. (2000) Design considerations for information Appliances. *In: Bergman, E. (Ed.) Information Appliances and beyond*, (pp.28 – 51). San Francisco; London: Morgan Kaufmann

Molich, R. www.dialogdesign.dk. (2002).

Murtagh, G.M. (2002) Seeing the "Rules": Preliminary Observations of Action, Interaction and Mobile Phone use. *In: Brown, B (2002) Wireless World: Social and Interactional Aspects of the Mobile Age; London: Springer, p81 - 91*

Nam, T. J., Lee, W.,(2003) Integrating Hardware and Software: Augmented Reality Based Prototyping Method for Digital Products. *CHI '03 extended abstracts on Human factors in computing systems*, Ft. Lauderdale, Florida, USA *April 05 – 10*

Nemeth, C.P. (2004). Human factors methods for design: making systems human-centred. London; Boca Raton, Fla, CRC Press

Norman, D.A (1998). The Invisible Computer: Why good products can fail, the personal computer is so complex and Information Appliances are the Solution. London; MIT Press

Norman, D. A. (2004). Emotional Design: why we love (or hate) everyday things. New York; Basic Books

Palen, L and Salzman, M (2002) Voice-Mail Diary Studies for Naturalistic Data Capture under Mobile Conditions. *Proc: 2002 ACM conference on Computer supported cooperative work*

Pering, C. (2002) Interaction Design Prototyping of Communicator Devices: Towards meeting the hardware – software challenge. *Interactions journal* 9(6) 36-46.

Raskin, J (2000). The humane interface: new directions for designing interactive systems. London; Boston, Mass, Addison-Wesley, 2000

Reeves, B. and Nass, C. (1998) The Media Equation:How people treat Computers, Television, and New Media Like Real People and Places. California, CSLI Publications

Rieman, J. (1993). The Diary Study: A Workplace-Oriented Research Tool to Guide Laboratory Efforts. *Proc: ACM INTERCHI'93Conference on Human Factors in Computing Systems, p.321-326*

Rubin, J. (1994) Handbook of Usability Testing; How to Plan, Design, and conduct Effective Tests. New York, John Wiley & Sons, Inc, p56

Sacher, H. and Loudon, G.H. (2002) Understanding the wireless interaction paradigm: From 3G technology to customer solutions, *Interactions, ACM. (2002) Volume IX.1, pp.17-23.*

Schrage, M. (2000). Serious play: how the world's best companies simulate to innovate; Boston; Harvard Business School press

Sharp, J (1998). Interaction design for electronic products using virtual simulations. PhD Thesis, Brunel University

Smith, C, D (1998) Transforming User-Centred Analysis into User Interface: The design of New-Generation Products *In: Wood, L E (1998) User Interface Design: Bridging the Gap from User Requirements to Design . - USA : CRC Press, p275 - 305*

Veggle, R. v., (2005) Where two sides of Ethnography Collide. *In: Design Issues*, *Vol 21, No 3*, Summer 2005

Vugt, H. v., and Markopoulos, P. (2003). Evaluating technologies in domestic contexts: extending diary techniques with field testing of prototypes. *In*: *Proceedings of the HCI International*, June 2003, Greece [In Press]

Villar, N., Lindsay, A. T., and Gellerson, H. (2005) Pin & Play & Perform. *In: Proceedings of the 2005 conference on New interfaces for musical expression*, Vancouver, Canada

Ware, C. (2004) Information Visualization: Perception for Design (2nd ed). San Francisco, Morgan Kaufmann

Woolley, A and Gill, S (2006) Information Ergonomics Lectures for Creative Prototyping, *In proc of*: *HCIEd.2006-1 inventivity: Teaching theory, design and innovation in HCI,* Limerick, Ireland

Simulation Model Interoperability in Support of Complex Organisation Design and Change

Richard. Weston, Min. Zhen, Aysin Rahimifard, Joseph Ajaefobi, Chenghua Ding, Alejandro Guerrero, Bilal Wahid and Tariq Masood
MSI Research Institute, Loughborough University
Loughborough, Leicestershire, LE11 3TU, UK.
E-mail: R.H.Weston@lboro.ac.uk

KEYWORDS

Manufacturing, decision support systems, dynamic modelling, combined simulation, process-oriented

ABSTRACT

Simulation Modelling has a key role to play in enabling decision making in dynamic manufacturing organisations. However in general the complexity levels involved necessitate multiple simulation models to be systematically developed and deployed. This paper describes a new systematic approach to creating coherent sets of simulation models that can interoperate to replicate and predict changing organisational behaviours.

INTRODUCTION

Manufacturing organisations are very complex yet need to function as dynamic systems, such that they remain competitive during their lifetime. One aspect of their complexity arises as understandings, knowledge and data (UKDs) about the organisation (and its business, managerial, technical and social structures and behaviours) is normally distributed amongst many knowledge holders. Hence to realise organisational change on any significant scale, **consultative decision making** is needed to

- conceive and agree upon improved ways of working
- resource and implement agreed changes.

It follows that constraints on consultative decision making will limit the quality and frequency of change decisions and impact negatively on the organisation's competitiveness.

Common change decision making in industry is centred on ad hoc meetings (involving persons with necessary influence, responsibilities and expertise) interspersed with periods during which responsible individuals consult with colleagues. Therefore current change decision making is often based upon accessing and processing distributed UKDs, but the processes used are typically very time consuming and ill structured. In some cases the time delays involved lead to 'solutions' to 'outdated problems', while in other cases pragmatic (non consultative) decisions are deemed necessary to facilitate responsiveness. The quality (fitness for purpose) of individual and group decisions made will first and foremost depend upon the quality of the personnel involved. However decision making qualities will also critically depend upon people availabilities and the time they can expend, the quality of UKDs they can access and the ease of that access.

With the foregoing observations in mind the present authors have (a) conceived and instrumented a new approach to structuring and enabling consultative decision making and (b) applied this approach within a number of small and large manufacturing organisations. Underlying research assumptions made (and being tested) are that suitable combinations of state of the art modelling frameworks, concepts and tools (including Enterprise Modelling, Dynamic Systems Modelling and Simulation Modelling) can be used to improve the quality and timeliness of organisation design and change decision making.

This paper considers in overview the role of Simulation Modelling (SM) in support of consultative decision making and reflects upon case study results.

USE OF MODELLING IN SUPPORT OF STRATEGY REALISATION

Weston et al (2006) explain that an overview of consultative decision making in manufacturing organisations can be gained through referencing Strategy Realisation (SR) activities. According to Mintzberg et al (1998), SR encompasses strategic thinking, strategy programming and strategy deployment. Weston et al (2006) also catalogue some popular business concepts with respect to different life phases of SR and explain how different classes of modelling technique can support decision making and action taking. Table 1 classifies types of organisation decision making that state of the art modelling techniques can naturally support. However, used on their own specific modelling technologies (including SM) can only provide limited support.

A 'COMPONENT-BASED', 'MIXED REALITY' APPROACH TO MODELLING ORGANISATIONS

The present authors have conceived and case tested the use of the Unified Organisation Modelling approach illustrated by Figure 1.

Table 1: Candidate Modelling Technologies – that support key aspects of strategy realisation

Purpose of Modelling	Nature of Modelling	Focus of Modelling	Example Threads of Modelling	Useful 'Business School' Concepts	Candidate Modelling Technologies
Opportunity Modelling	abstract structural & behavioural modelling, of wide scope with a medium to long-term focus	modelling causality between the ME & its environments to identify strategic opportunities	* Analyse market structure, requirements & opportunities * Analyse product portfolio opportunities * Analyse competitor strengths & weaknesses * Conduct internal capability analysis * Identify candidate strategic intents	* Scenario Planning * Game Theory * Porters Forces * Organisational Structure (Mitzberg & Walters) * Supply Chain Analysis * Process Classifications * Competitor Analysis	* Dynamic systems modelling * Causal loop modelling * Dependency & constraints modelling * Simulation based on numerical integration
Requirements Modelling	primarily mid-level abstractions of structural aspects, of medium to wide scope with a medium to long-term focus	modelling of needed process oriented organisational forms that the ME can use to compete for identified opportunities	* Analyse business process networks relative to candidate strategic intents * Analyse supply chain alternatives * Analyse work organisation alternatives under financial &resource constraints * Analyse & short list viable production strategies * Identify candidate business models	* Porters Forces * Organisational Structure * Process Classifications * Benchmarking * Process configurations (MTS-MTO-ETO etc) * Product Structure * VAT Plants * Postponement * Lean/JIT/Agile	* Enterprise & process modelling * Various types of simulation modelling * Value stream analysis * Cost modelling * Decision network modelling
System Solutions Modelling	primarily mid to low level abstractions of structural & behavioural aspects, of narrow to medium scope with a short to medium-term focus	modelling to inform the resourcing of process-oriented roles with candidate human & technical systems & analysing & predicting their behaviours when subject to work pattern dynamics	* Identify process oriented roles amongst viable Business Process networks * Match viable candidate human & technical resource systems to identified roles * Analyse competitor strengths & weaknesses * Exercise resourced roles with historical and predicted work patterns * Select suitable work structures via dynamic analysis of short listed business & production strategies	* Process classifications * Postponement * Lean * JIT- Kanban * CONWIP * Agile * Competency models * Team working theory * Scheduling techniques * Flexibility theory	* Enterprise & process modelling * Discrete event simulation * Value stream analysis * Cost modelling * Role modelling * Resource capability modelling * Work pattern modelling * Exception modelling

Underlying assumptions being tested are that: (1) 'reusable components' (both modelled and real) of organisations can be 'configured' into 'interoperating systems of mixed reality components', such that these systems can realise changing organisational requirements (including ongoing change in customer demand for new and existing products and services); (2) models of real systems components can be deployed (with sufficient quality and utility) by combining the use of state of the art modelling techniques to capture and exercise UKDs in support of timely and effective consultative decision making; (3) potential organisational benefits arising from using mixed reality component based modelling environments can significantly outweigh the cost of their continued deployment.

INTEGRATING CONCEPTS

The authors and their colleagues have adopted the use of existing modelling concepts and technologies and as required have conceived and deployed new integrating modelling concepts.

Public domain Enterprise Modelling (EM) techniques were observed to usefully provide means of handling organisational complexity, by offering modelling concepts to decompose (general and specific) process networks into their component process segments. Also existing EM techniques were observed to provide means of documenting and visualising associated flows of activities, material, information, controls and so forth. Thereby UKDs distributed amongst personnel concerned with 'operational', 'tactical', 'strategic' and 'infrastructural' processes of any organisation can be modelled in a visual, reusable fashion; so as to formally specify what needs to be done by the organisation over given timeframes and how various decisions and actions carried out can causally impact on other process segments of the organisation. Also observed were various complementary process, product and resource modelling techniques which can be used to attach specific structural and parametric data to Enterprise Models (EMs) so as to provide a 'big picture' of the current organisations 'configuration'. Such a 'big picture' provides a framework for positioning various kinds of UKD and proved useful to decision makers in the case organisations modelled. However such a developed EM naturally only encodes relatively enduring properties of organisational entities and relationships between those entities.

To enhance the utility of EMs and their connectivity with dynamic (time dependent) models of selected enterprise components the present authors conceived and developed the use of 'role' and 'dynamic producer unit' (DPU) modelling concepts.

In general 'process and organisation designers' need some means of determining sets of 'roles' that must be resourced (by suitable systems comprising human and/or technical resource elements) to realise the various ordered sets of activities that comprise a specific process network. To satisfy this need the present authors deploy decomposition principles of EM techniques, so as to identify 'organised sets of process segments' (i.e. 'component building blocks' of process networks) which can be treated as being equivalent to 'organised sets of roles'. Of course in real organisations various feasible decompositions may be determined and this leads to the identification of alternative role sets, for which alternate resource systems may be assigned a responsibility. A key advantage of using an enterprise model to determine viable role sets is that naturally previous activity, material, information and control flows related to each process segment will already have been explicitly specified. This phenomenon is used by the authors to explicitly model 'role requirements' for specific process network cases. Further, by understanding the nature of the activities, and activity relationships associated with each role, it is natural to explicitly attach to each 'role requirement' model, explicit descriptions of 'competency requirements' needed to realise each role. To operationalise the use of explicit 'role requirements' definitions during consultative decision making, the present authors have also developed the use of complementary models of 'potential roles' that candidate resource systems could play within a specific process network. By using common modelling concepts to explicitly describe 'required' and 'potential' roles, suitable candidate human and technical resource systems can be systematically identified and short-listed as viable role holders.

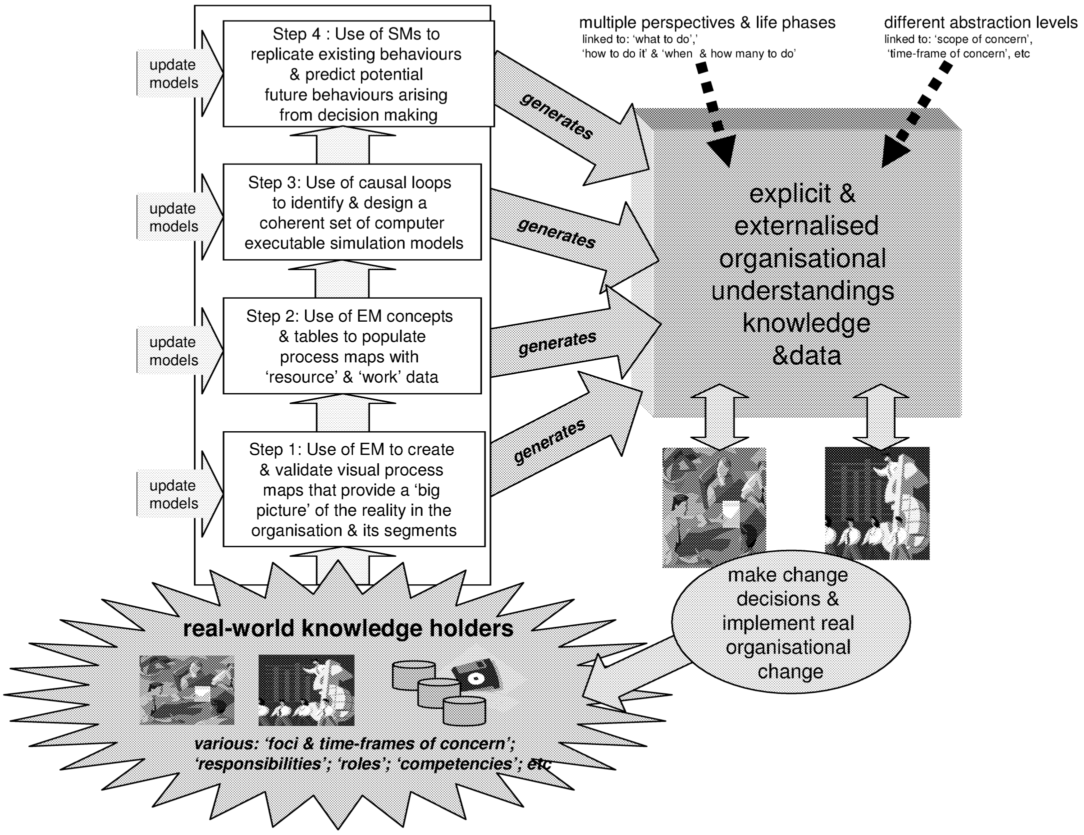

Step 4 : Use of SMs to replicate existing behaviours & predict potential future behaviours arising from decision making
Step 3: Use of causal loops to identify & design a coherent set of computer executable simulation models
Step 2: Use of EM concepts & tables to populate process maps with 'resource' & 'work' data
Step 1: Use of EM to create & validate visual process maps that provide a 'big picture' of the reality in the organisation & its segments
update models
update models
update models
update models
generates
generates
generates
generates
multiple perspectives & life phases
linked to: 'what to do',' 'how to do it' & 'when & how many to do'
different abstraction levels
linked to: 'scope of concern', 'time-frame of concern', etc
explicit & externalised organisational understandings knowledge &data
make change decisions & implement real organisational change
real-world knowledge holders
various: 'foci & time-frames of concern'; 'responsibilities'; 'roles'; 'competencies'; etc

Figure 1 Overview of systematic modelling approach- leading to the development & deployment of coherent simulation models

The DPU concept was conceived as a means of achieving coherent abstract descriptions of common reusable components (or building blocks) of manufacturing organisations. Here it was assumed that (1) DPUs will function individually, as a holder of one or more assigned roles and (2) configurations of multiple DPUs will interoperate so as to function collectively as holders of one or more higher level (more abstract) roles (i.e. roles composed of lower level roles). In real manufacturing organisations, actual building blocks comprise various systems of people, production machines and computers. These common building blocks (or systems of resources) are typically configured into various systematically operating groups (via the imposition of organising structures and parametric data) so that they function and behave as required in a specific workplace and under specified sets of workload conditions. It follows that configured resource systems need to possess specified competencies, behaviours and levels of performance to realise all needed instances of process segments to which they are assigned. Example stereotypical resource systems (or DPUs) include workgroups, teams of people, production cells, production lines, workshops, departments, business units, companies and partnership enterprises. Hence a research assumption being tested is that all such types of organisational unit can be usefully modelled using role and DPU modelling ideas as a means of treating them as configurable, reusable and interoperable components of complex organisations. As illustrated by Figure 2 therefore it has been assumed that physical and logical configurations of DPUs (whether they actually comprise people, machines and/or computers) can all be usefully characterised in terms of their:

- *Relatively enduring DPU functionality* – expressed in terms of 'functional competencies', including for example competencies to assemble product X, process orders of type Y and design products of type Z. (Here the term 'competencies' is considered to encompass human systems oriented competencies and technical (machine and computer) system capabilities, bearing in mind that many enterprise activities can be realised by either people, machines, computers or organised combinations of these active resource types.)

- *Relatively enduring DPU structures* – expressed in terms of activity, information, control and material flows that are linked to role assignments and descriptions of needed interactions between roles
- *DPU dynamic characters* – expressed in terms of performance levels (e.g. lead-times, rate of value addition and costs consumed), behaviours (e.g. availability, reliability, change capability and operational flexibility) and relevant cultural concerns (e.g. level of workforce motivation and influential cultural values).

In case organisations considered thus far, by modelling stereotypical DPUs as potential holders of roles, significant benefit has been observed; this has enabled the design and explicit specification of systematic methods for modelling organisations, their change requirements and impacts of change types on organisations, and has provided a formal basis for instrumenting new ways of externalising and reusing UKDs.

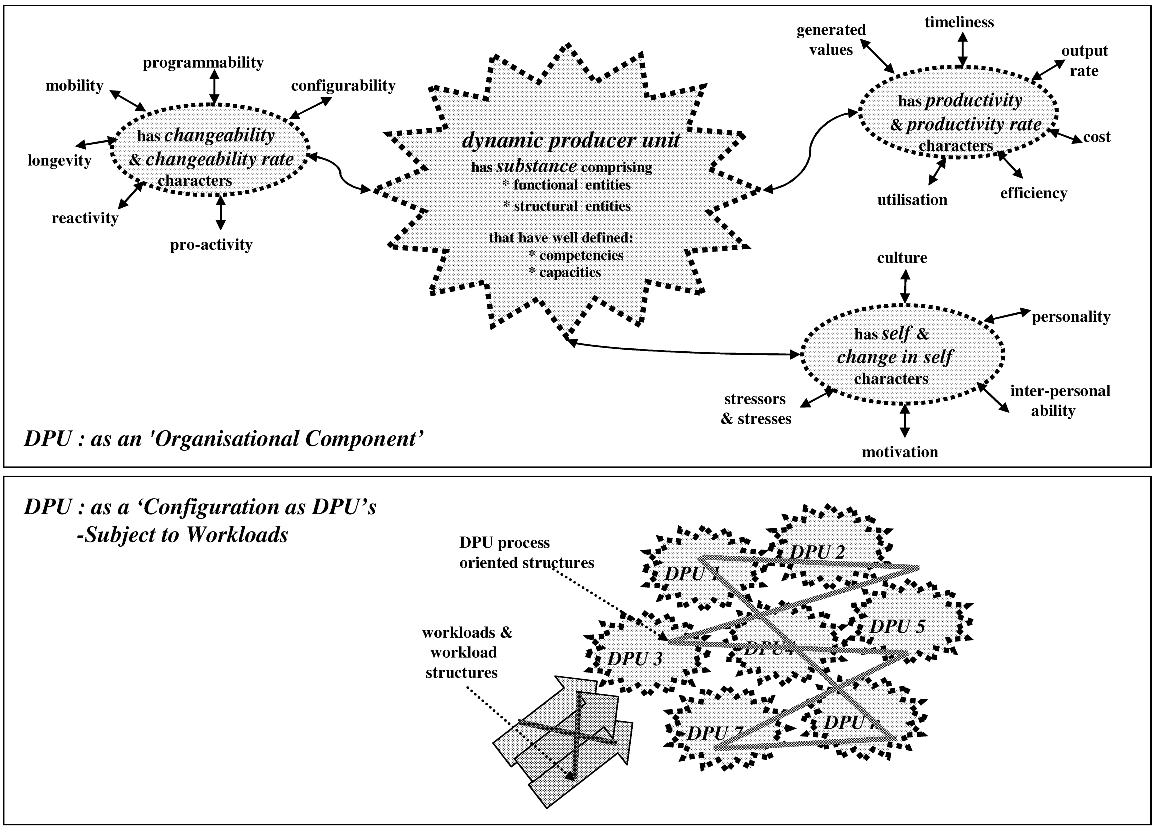


Figure 2 The Dynamic Producer Unit (DPU) Concept

COHERENT MODELLING OF HUMAN AND TECHNICAL RESOURCE SYSTEMS

Role modelling, work pattern modelling and DPU concept development has centred on enabling a two stage process of (I) short-listing viable candidate resource systems and (II) selecting from amongst viable candidates by predicting and comparing their performance and behaviours under varying workload conditions.

During stage (I), DPU characterisations of candidate configurations of resource elements are compared in terms of the relatively enduring functionality (i.e. competencies) they can bring to bear on specific workplace roles; thereby providing a first stage systematic basis for selecting between candidates and drawing up a short list of viable resource systems. To explicitly systemise resource system selection during stage (I), a previously captured Enterprise Model describing the case organisation (and its current process network) is analysed, assuming that it comprises 'process segments' (i.e. organised groupings of enterprise activities) that collectively specify a natural decomposition of a specific case of 'required roles' and 'dependencies between required roles'. The approach of considering 'process segments' as being 'possible roles' which can be played by 'alternative candidate resources' has provided significant flexibility with respect to organisation design and change, yet can formally specify key aspects of roles and role assignments. The approach has also provided useful explicit descriptions of dependencies between roles which can later be referenced during resource system implementation as explicit structural descriptions of control information, material and data flows associated with different configurations of DPUs and their varying assignment to roles and specific instances of roles.

During stage (II), dynamic systems analysis (based on the combined use of causal loop modelling and simulation modelling) is carried out to select between short-listed candidate resource systems on grounds of their ability to (1) perform given work patterns and (2) behave appropriately, so as to befit their specific work environment short, medium and long term.

Causal loop modelling is used to understand in qualitative terms how causal and temporal impacts propagate through complex organisations as dynamic patterns of work (e.g. works orders, projects, etc) are assigned to ‘process segments’, ‘roles’ and ‘role holders’. This has proven effective as a basis for specifying the purpose, scope and focus of multiple simulation models that individually can support resource system design and change decision making and collectively can replicate and predict performances and behaviours in the wider case organisation. This naturally leads on to (a) the design of simulation models and simulation modelling experiments and (b) the ability to realise interoperation between simulation models.

ILLUSTRATIVE CASE

Because of space constraints, this paper will only illustrate in outline how the concepts reported in this paper have been beneficially applied; so as to deploy simulation modelling in support of complex decision-making in a case study manufacturing organisation. In this case study the method of externalising distributed UKDs illustrated by Figure 2 was deployed to provide a coherent set of enterprise, causal loop and simulation models. Following model validation involving extensive discussion with knowledge holders, the developed set of models explicitly documented key characteristics of the current configuration and current reachable states of the case organisation. Figure 3 illustrates examples of some of the current state models created; where these models took various forms including: graphical models of relatively enduring entities and entity relationships; tabulated models related to (process, resource and product) structures, parameters and data; graphical models of causal and temporal impacts linking organisational variables; and various computer executable models that are exercised by simulation and workflow management tools.

The case company employs circa 50 people to make high quality pine furniture in response to orders received mainly from furniture stockists. Circa 350 product variants are made, each of which can have a number of colour finishes. Many of the case company problems revolved around their product dynamics; because the mix and volume of products ordered during any given planning window has (and likely will continue to) varied very significantly. Therefore key issues were to maintain competitive product quality, lead-times and cost despite the product dynamics and constraints arising from a need to maintain a sufficiently competent and change capable set of human and technical resources. The company had also experimented by implementing various organisational changes, alternative business and manufacturing policies and rules, new business systems and had sought to minimise waste and cost, whilst coping with human resource change and maintaining flexibility where and when required. However inevitably it faced significant complexity issues and previously had no analytical basis for change decision making.

In collaboration with case company personnel, the university team (mainly comprising the present authors) has successfully used the current configuration and state models

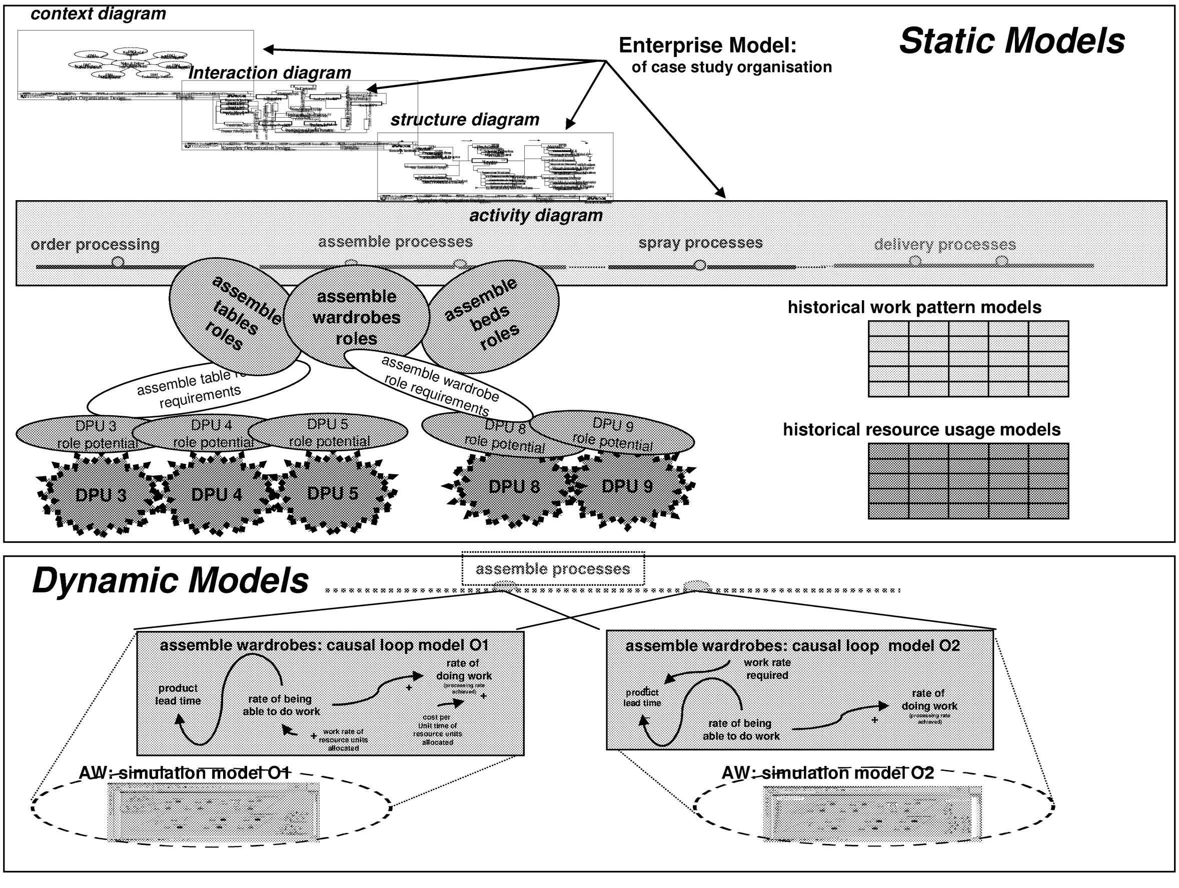


Figure 3 Case Study Illustration of the Modelling Methodology: to create various simulation models of individual process segments

(illustrated by Figure 3) along with various future configuration and state models to provide analytical decision making support. On an ongoing basis this is improving the competitiveness of the case organisation by minimising time loss and the loss of significant investments in change (that previously had resulted in poor performance because of making ill advised change decisions). In a number of related modelling studies the authors have recommended (1) localised improvements to specific process segments of prime concern to the case organisation and (2) recommended improved business and manufacturing policies that span multiple process segments.

REFLECTION AND CONCLUSIONS

This study has observed key roles for simulation modelling in support of complex organisation design and change. However it has also observed practical constraints on the use of single simulation models, in that they can only either model (1) the whole organisation in a simplistic manner or (2) segments of the organisation in detail, based on the assumption that segmented models can usefully be modelled in isolation.

The purposes of the modelling concepts and approaches reported in this paper are to:

(a) capture and operationalise UKDs distributed mainly amongst human knowledge holders in complex organisations.
(b) enable unified use of enterprise models, causal loops and simulation and workflow modelling to understand and analyse specific organisational dynamics.
(c) provide an explicitly defined foundation for model unification and simulation model interoperation.

Early findings when modelling a number of small and large manufacturing organisations have been very encouraging. Although more extensive testing is required in respect of (c), the use of process network, role, DPU and resource systems (competency and performance level) modelling concepts (informed by causal loop modelling) has provided an enhanced basis for creating coherent simulation models. As illustrated by Figure 4, the developed modelling methodology results in experimental simulation models that share common semantics about a specific and complex organisation. Further key separations related to structural aspects of these simulation models facilitate both decoupling and flexible integration of ‘process’, ‘resource’ and ‘work pattern’ aspects. Therefore in theory the modelling structures, concepts and techniques researched can usefully input to ontological developments related to complex organisation design and change.

REFERENCES

Ajaefobi J.O. 2004. “Human systems modelling in support of enhanced process realisation”. PhD Thesis, Loughborough University, Leics., UK.

Chatha K.A. 2004. “Multi-Process Modelling Approach to Complex Organisation Design”. Ph.D. thesis, Wolfson School of Mechanical and Manufacturing Engineering, Loughborough University, Leics., UK.

Mintzberg, H., Ahlstrand, B., and Lampel, J., 1998, “Strategic safari: the complete guide through the wilds of strategic management”, Prentice Hall, UK, ISBN: 0273656368.

Monfared, R.P.; West, A.A.; Harrison, R.; Weston, R. H. 2002. “An implementation of the business process modelling approach in the automotive industry”, Proc. of the Institution of Mech. Engineers, Part B: J .of Engineering Manufacture, v 216, n 11, (2002), 1413-1428.

Weston, R H, Guerrero, A and Chatha, K.A. 2006. “Process Classes Deployed in Manufacturing Enterprises”, accepted for publication in IJCIM.

DECISION BASED SIMULATION

THE APPLICATION PLA FOR CREATION SIMULATION MODELS FOR DECISION MAKING

H.Pranevicius
V.Pilkauskas
D. Makackas
Kaunas University of Technology
Studentu 50
LT-51368 Kaunas, Lithuania
E-mail: hepran@if.ktu.lt, vytpilk@ktu.lt, damaka@if.ktu.lt

KEYWORDS
Business process modeling, piece linear modeling, decision making, oil terminal.

ABSTRACT

Business process modeling and piece linear aggregate (PLA) formalism are used for creation simulation model for decision making in oil terminal. It is presented background of pieced linear aggregates and its relation with business process models notations. Library object oriented simulation system for creation PLA models are used.

INTRODUCTION

A decision marker in an enterprise is expected to make a decision that has a positive effect on its future. Enterprise information systems (EIS) should support that activity. An information system is one that supports decision-making by providing past and future data to a client program. The client program aids the decision maker by incorporating analysis and planning algorithms that assess the value of alternate decisions to be made now or at points in the near future (Figure 1).

Data about the past are received from a database while a forecast about the future is performed using simulation (Wiederhold 2000). While using a simulation model to forecast the future it is important to evaluate a current system state. Information stored in the database cannot fully evaluate the current state. Attributes that define the current state are corrected at different time moments. In order to obtain attribute values of the system state at the current time moment, the simulation model is used to extrapolate data.

Decision making using simulation is a dynamic goal-oriented decision making and is used in systems that constantly in time make decisions in order to optimize system key performance characteristics (Dalal et al. 2003). Decision making systems, which use simulation, are characterized with the following parameters:

- Strategy. Which strategy to use for decision making during simulation of the future?
- Duration. How long to run simulation of the future?
- Number of iterations. How many to repeat simulation of the future to evaluate each decision making alternative?
- Heuristics. Which heuristics to use to evaluate system key performance characteristics that are obtained during simulation of the future?

The simulation model is used to evaluate system characteristics for each possible decision alternative. This permits to choose the best decision alternative.

Simulation models can provide the most accurate and insightful means to analyze and predict the performance measures of business processes. Simulation is a powerful tool for allowing designers imagines new systems and enabling them to both quantify and observe behavior. In the past few years, several new software tools have been developed specifically for modeling business processes (Kirikova 2005; Pranevicius 2003). Most of these tools define business processes using graphical symbols or objects, with individual process activities depicted as a series of boxes and arrows. Special characteristics of each process or activity may then be attached as attributes of the process. Business process simulation software tools can be placed into three major categories:

- Flow diagramming-based simulation tools,
- System dynamics-based simulation tools,
- Discrete event-based simulation tools.

The most capable and powerful tools for business process

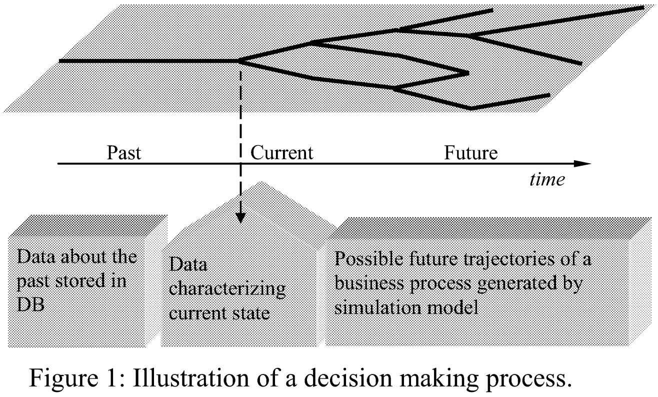

Past
Current
Future
time
Data about the past stored in DB
Data characterizing current state
Possible future trajectories of a business process generated by simulation model

Figure 1: Illustration of a decision making process.

simulation are the discrete event-driven simulation products. These tools provide modeling of entity flows with animation capabilities that allow the user to see how flow objects are routed through the system. Some of these tools even provide object-oriented and hierarchical modeling, which simplifies development of large business process models. Business simulation can aid in decision-making that project complex what-if scenarios, offering a reliable way to evaluate the likely effects of different decisions and variables.

In this paper we present a piece-linear aggregate (PLA) (Pranevicius et al. 1994) formalism for creation of dynamical models of business processes. Motivations to use the formal technique are:

- It permits to prepare a formal description of analyzed system having one meaning;
- Properties of model may be analyzed using mathematical proof techniques;
- Formal description approach acts as a is a theoretical background while developing software tools for computerized analysis (validation, verification, simulation) of formal specifications.

BUSINESS PROCESS MODELING

According to the Business Process Reengineering literature a business process can be viewed as a set of partially ordered activities intended to reach a goal. The above definition, besides the notion of business process, introduces the following notions:

- Goal, or objective of the process,
- Activity, often called task, and,
- Time - an axis, which the partial order refers to.

The time can be regarded as being absolute or relative, discrete or continues, internal or external.

The notion of objective (goal) presumes that at any moment of the interval of time when the process exits we can tell whether the process objective is achieved or not. If it is not, we would like to be able to tell how far it is to the process goal. This leads us to the notion of the process state. The state can be final, if the objective has been reached, or intermediate, otherwise.

The notion of state helps to define the notion to activity. An activity is viewed as an action aimed at changing the process state in a special way. The definition of activity above is based on the notion of change in the process state. The notion of event in the process lifetime means a moment of time when process state changes. Each completed activity results in an event.

The notion of business process, which was discussed above, refers to the specific process that evolves in time. Processes that have similar goals and similar patterns of behavior (the same kind of activities) can be united by the notion of process type. The above notions allow considering a business process as a dynamic system, which moves in the space of all possible process states until it reaches the final state (the objective).

In this paper Business Process Modelling Notation (BPMN) (Kirikova 2005) will be used. The goal of BPMN is to provide a business process modeling notation that is readily usable by business analysts, technical developers and business people that manage and monitor these processes. BPMN allows to create a Business Process Diagram which represent the activities of the business process and the flow controls that define the order in which they are performed. Business process is modeled by four irreducible concepts: Actors, Activities, Events and States. BPMN specifies four categories of objects:

- Flow objects are made up of events, activities and gateways;
- Connecting objects are: a sequence flow (simple transition), a message flow (a transition guarded by a message event) or an association (that is used to associate data, text, to flow objects);
- Swimlanes represent participants;
- Artifacts are mainly data objects. Data objects are typed and represent the input and output of activities.

PIECE-LINEAR AGGREGATES

Most of the existing semantic models, language and logics for describing and reasoning about timing-based systems implicitly view an execution as an alternating sequence of instantaneous "discrete" actions and "continuous" phases during which time advances. To each system described in any of these formalisms one can associate a transition system or automaton consisting of a set of subsets: a subset of initial states, a subset of discrete actions, a subset of discrete steps $s' \xrightarrow{a} s$ asserting that "from state s' the system can instantaneously move to state s via the occurrence of the discrete action a', and, finally, a subset of time-passage steps $s' \xrightarrow{d} s$ asserting that "from state s' the system can move to state s during a positive amount of time d in which no discrete action occurs". These transition systems provide a very abstract view of the behavior of the original system in which many aspects, such as the number of parallel components, the communication between these components, the way in which a system evolves during the continuous phases, etc., are no longer represented.

PLA is a special case of automaton models. In the application of the PLA approach for system specification, the system is represented as a set of interacting piece-linear aggregates. The PLA is taken as an object defined by a set of states Z, input signals X, and output signals Y. A behavior of an aggregate is considered in a set of time moments $t \in T$. State $z \in Z$, input signals $x \in X$, and output signals $y \in Y$ are considered to be time functions. Transition and output operators, *H* and *G* correspondingly, must be known as well.

The state $z \in Z$ of the piece-linear aggregate is the same as a state of a piece-linear Markov process, i.e.: $z(t) = (\upsilon(t),\ z_\upsilon(t))$, where $\upsilon(t)$ is a discrete state component taking values on a countable set of values; and $z_\upsilon(t)$ is a continuous component comprising of $z_{\upsilon 1}(t), z_{\upsilon 2}(t), \ldots, z_{\upsilon k}(t)$ coordinates.

When there are no inputs, an aggregate the state changes as follows:

$$\upsilon(t) = \text{const}\,,\ \frac{dz_\upsilon(t)}{dt} = -\alpha_\upsilon\,,$$

where $\alpha_\upsilon = (\alpha_{\upsilon 1}, \alpha_{\upsilon 2}, \ldots, \alpha_{\upsilon k})$ is a constant vector.

The state of aggregate can change in two cases only: when an input signal arrives to the aggregate or when a continuous component acquires a definite value. The theoretical basis of piece-linear aggregates is their representation as a piece-linear Markov processes.

Continuous coordinates, which are used in PLA, define time moments when internal events occur. The aggregate state $z(t_m)$ can be changed only at discrete time moments $t_m, m = 1,2,3,\ldots,$ remaining fixed in every interval $[t_m, t_{m+1}], m = 0,1,2,\ldots,$ where t_0 – the initial moment of system behavior. When the system state system is known $z(t_m), m = 0,1,2,\cdots,$ the moment t_{m+1} of the following event is determined by a moment of input signal arrival to the aggregate or by the equation:

$$t_{m+1} = \min\{w(e_i'', t_m)\},\ 1 \le i \le f\ .$$

A class of the next event e_{m+1} is specified by input signal, if it arrives at the time moment t_{m+1} or is determined by control coordinate having the minimum value at the moment t_m, i.e. when the coordinate $w(e_i'', t_m)$ becomes minimal, $e_{m+1} \in E''$.

The new aggregate state is stated by *H* operator.

$$z(t_{m+1}) = H[z(t_m), e_i],\ e_i \in E' \cup E''\,,$$

where E' and E'' mean subsets of external and internal events correspondingly.

Output signals y_i from the set of output signals $Y = \{y_1, y_2, \ldots y_m\}$, can be generated by an aggregate only at moments of events from the subsets E' and E''. The operator *G* determines the content of the output signals:

$$y = G[z(t_m), e_i],\ e_i \in E' \cup E'', y \in Y\ .$$

RELATION BETWEEN CONCEPTS OF BUSINESS PROCESS MODELING AND PLA

Let us denote a set of concepts of a business process T_{BP} = {Goal; Activity: Time; Change; Event; Chronicle} and a set of concepts of *PLA* model T_{PLA} = {final state: $z(t_m)$; Operation: O_i; time $= t_m$; Transition operator: $H(e)$; Sets of events: E', E''; Trajectory: $z(t_1), e_1'', z(t_2), e_2'', \ldots, z(t_m)\}$.

Assertion: Relation between concepts of business process and concepts of PLA approach can be described by a partial injective function *R* over $T_{BP} \times T_{PLA}$, i.e.,

$$\forall t_1 : T_{BP}; t_2, t_3 : T_{PLA} \cdot (t_1 R t_2 \wedge t_1 R t_3) \Rightarrow t_2 = t_3) \wedge$$

$$\forall t_1, t_3 : T_{BP}; t_2 : T_{PLA} \cdot (t_2 R^{-1} t_1 \wedge t_2 R^{-1} t_3) \Rightarrow t_1 = t_3)$$

Graphically function *R* is presented in Figure 2

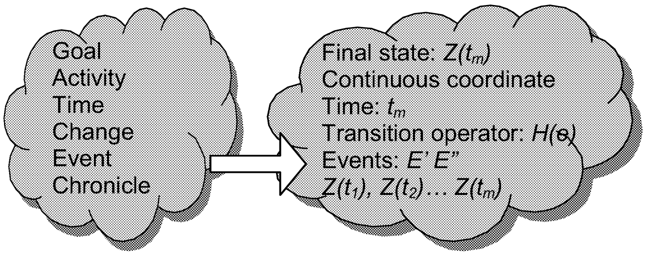


Figure 2: Relation between of notations of business process and PLA models

LIBRARY OF OBJECT-ORIENTED SIMULATION SYSTEM

Object-oriented library for development of simulation models of systems described by PLA specifications is created (Pranevicius and Pilkauskas 2002). The library is made of packages of internal and external events. Four simulation model algorithms of an aggregate system are implemented in the package of external events:

- Creation of aggregate modules and their connection into the closed aggregate sys-tem;
- Setup of the initial state of the aggregate system;
- Generation an external event and its putting to a queue of external events of the aggregate system;
- Selection of an external event and its passing for processing in a corresponding aggregate.

Two algorithms of simulation models of the aggregate system are implemented in the package of internal events:

- Generation an internal event and its putting to an ordered queue of internal events of the aggregate system with respect to time moments of event occurrences;
- Selection of an external event and its passing for processing in a corresponding aggregate.

This library is implemented using JAVA and C# programming languages.

A SYSTEM FOR CREATION A SCHEDULE OF TANKER LOADING AT OIL TERMINAL

In this section a description of the system for creation a schedule of tanker loading at oil terminal is presented. As depicted in Figure 3 the system is made of:

- DMWebApp - web application for control of simulation model and presenting results of simulation experiments
- PortalEnginer- integrating layer;
- SimWebService - web service of terminal simulation model;
- TerminalSimModel- terminal simulation model;
- EIS_DB - terminal information system database.

Terminal model simulates oil deliver to the terminal by a train. Duration of loading from train tanks to reservoir depends on oil temperature and this dependency is known. Three platforms are used for loading of oil in the terminal. Arrived trains are placed to a queue when all platforms are occupied. Trains are served according FIFO servicing strategy. The terminal has two embankments for loading of tankers. Business model of train unloading in oil terminal is presented in Figure 4.

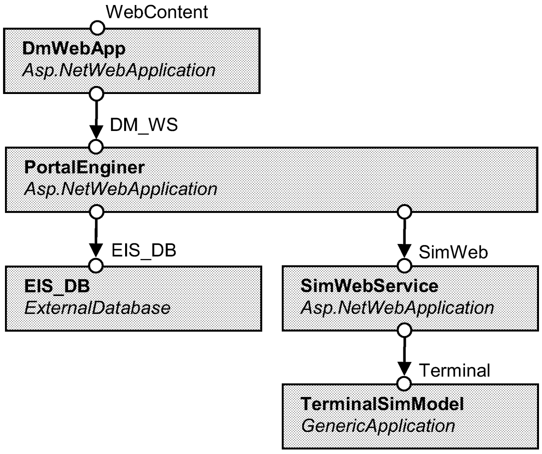


Figure 3: Diagram of the system for creation a schedule of tanker loading at oil terminal.

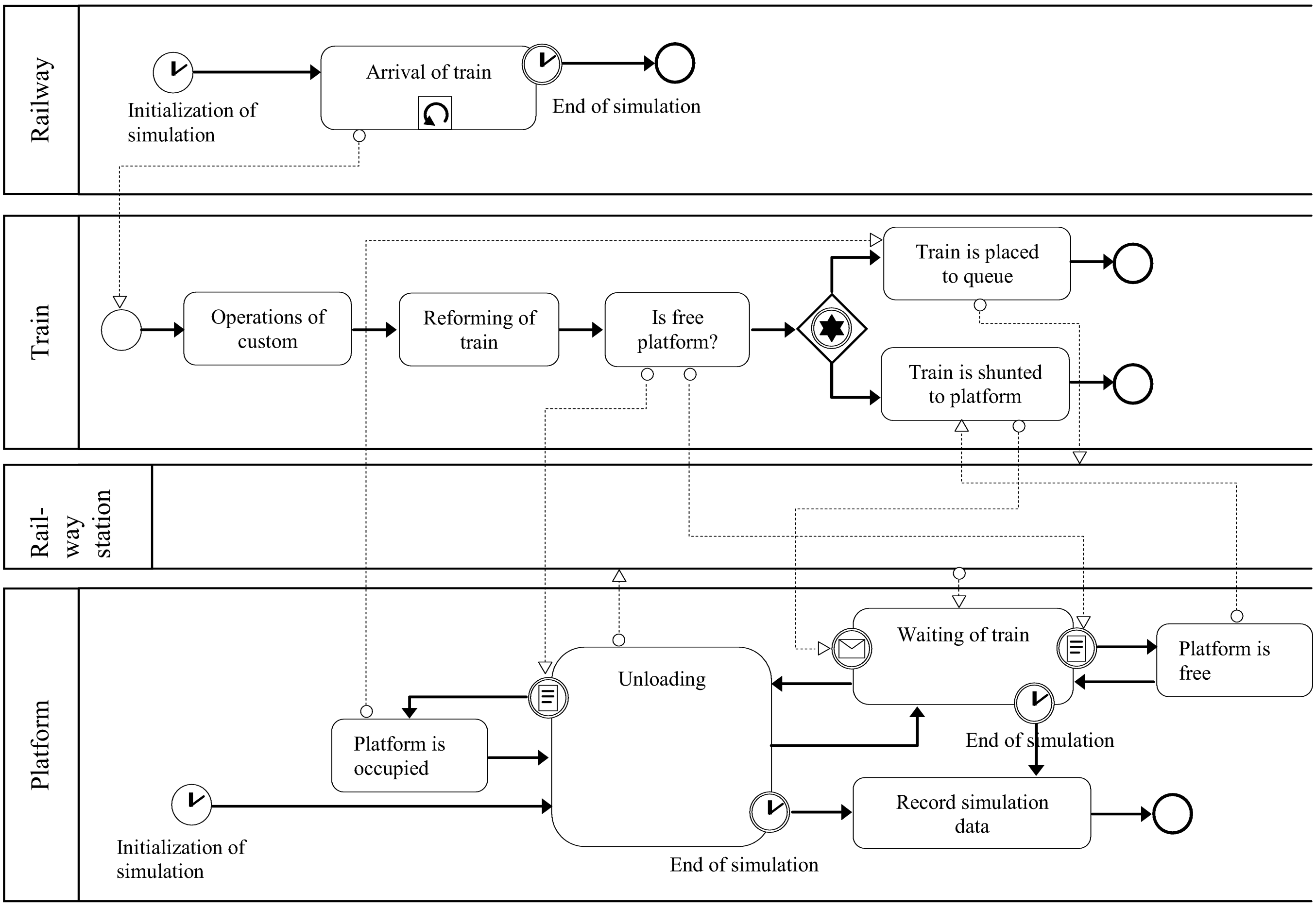


Figure 4: Business model of train loading in oil terminal.

Simulation results are presented in a graphical view, as depicted in Figure 5, after an execution of simulation experiment with one of a schedule variant of tanker loading.

The digits in Figure 5 mean:

1 - Accumulated oil amount at oil terminal before a simulation start moment;

2 - Amount of oil products to be loaded to oil reservoirs till a tanker loading start moment;

3 - Amount of oil products to be loaded to tanker reservoirs directly from wagons;

Also conflict situations (which are depicted by letters in Figure 5) are estimated:

A. Tankers will lack a time to maneuver at embankments;

B. It will be impossible to load two tankers from wagons;

C. It will be a lack of oil products to load tankers;

D. Wagons with oil products will arrive too late;

It is necessary to correct a schedule of tankers loading to remove observed conflict situations. Then, the simulation experiment can be performed again with a new loading schedule.

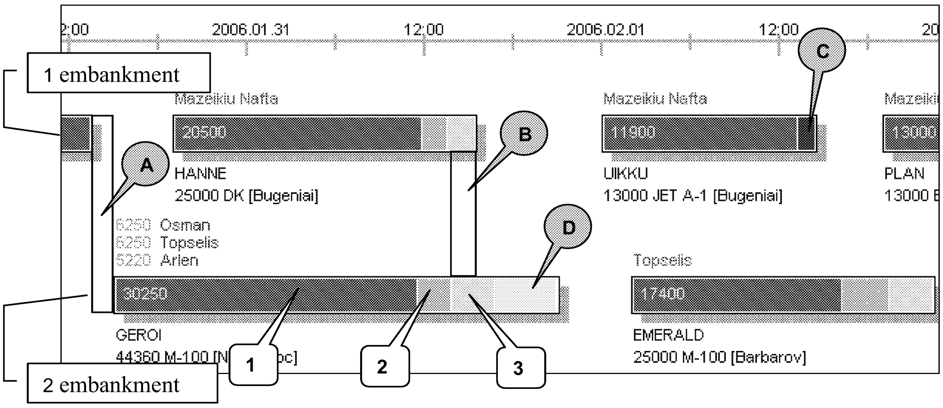


Figure 5 Example of simulation model result

CONCLUSIONS AND FUTURE WORK

Most of simulation model development tools are not directly use models of business processes. Further, we plan to develop a technique for creation simulation models of business processes that would include: Business Process Modeling Notation and formalization of business processes using PLA. Model Driven Engineering (MDE) principles are planned to use for automated generation of simulation model program code from BPMN. Meta-models of source and target models will be defined. This will permit to create model transformation from a business process model to PLA model.

REFERENCES

Wiederhold G. 2000. “Information system that really support decision-making.” *Journal of Intelegent Information System*, vol. 14, 85–94.

Dalal M.; B. Groel; and A. Prieditis. 2003. “Real-time decision making using simulation”. In *Proc. 2003 Winter Simulation Conf.,* S. Chick, P. J. Sanchez, D. Ferrin, and D. J. Morrice (Eds.). 1456-1464.

Kirikova M. and J. Makna. 2005. “Renaissance of business process modelling”. In *Information Systems Development Advances in Theory, Practice, and Education,* O. Vasilecas, A. Caplinskas, G. Wojtkowski, W. Wojtkowski, and J. Zupancic (Eds.). Springer, 403 - 414.

Pranevicius H. 2003. “Formalization and simulation of business process”. In *Int. Conf. Modelling and Simulation of Business Systems*. 198-202.

Pranevicius H.: V. Pilkauskas; and A. Chmieliauskas. 1994. *Aggregate Approach for Specification and Analsysis of Computer Network Protocols*. Kaunas, Technologija.

Pranevicius H. and V. Pilkauskas 2002. “Object Oriented Library for Developing Simulation Models Specified by Aggregate Approach”. In *Int. Workshops on Harbour, Maritime and Multimodal Logistics Modelling & Simulation. Modelling & Applied Simulation*, 122-127.

BIOGRAPHY

HENRIKAS PRANEVICIUS Professor, Kaunas University of Technology, Head of Business Informatics Department. Habilituated doctor of Technical Sciences at Ryga Electronic and Computer Technik Institute, 1984. Doctor degree in Kaunas Politechnical institute at 1970. Area of research activity: formal specification, validation and simulation of distributed systems including telecommunication and logistic systems. The theoretical background of investigation is piece-linear aggregate formalism, which permits to use the single formal specification for models development both for performance and behaviour analysis.

THE SIMULATION OF THE ECONOMIC EFFECT OF POWER SYSTEM STRUCTURE INCLUDING RENEWABLE SOURCES OF ENERGY

Eugeniusz M. Sroczan
Institute of Electric Power Engineering
Poznan University of Technology
ul. Piotrowo 3A
60-965 Poznan, Poland
E-mail: sroczan@put.poznan.pl

KEYWORDS

Energy management, computer aided analysis, decision support system, interactive simulation, optimization.

ABSTRACT

The optimal load of the power system (PS) sources depends on the structure of committing sources, that might affect the generation costs in the given power network. The expected costs of energy for the given wholesale providers and power plants are calculated using some simulation procedures, assuming the varied structure of sources – types of power plants. In the short time period the results support the real time operation of PS. Over a longer time they enable the definition of the optimal policy of changing or modernizing the power structure. Operation of an example set of the discussed power plants, including all types of primary energy sources and converters, including also the pumped storage hydroelectric plant (PSHP) and wind power plants (WPP), is described in accordance with the balance of the load in the PS. Upon the base of the developed simulator, the mode of operation the sources load is based on the minimization of the cost of generated energy. The structure of the simulator (Sroczan 2005) consists of some procedures which simulate the generation policy and procedures checking the economic quality of balancing the generation and demand of electric power and energy in the discussed PS.

INTRODUCTION

The rules of the wholesale energy market, which describe realization of the competition among producers as well as distributor companies, should assure the interests of energy consumers. A Local Energy Market (LEM) allows the producers and providers to sell both the electric energy and heat to the consumers near the sources. The essential problem of LEM is to balance the energy production with consumers' demands, with regard to the technical and economic boundaries and the power flow balanced in the local power system, expecting a varied set of energy sources.

The directives of the European Parliament and of the Council of European Union assume that in the near future the generation of energy based on renewable sources in EC will consist of approximately 7,5-20 %.

The main aim of this paper is to simulate the effect of the structure of the electric power sources set in order to minimize the costs of energy. The additional constraints are in the form of varied weather and different customer behaviors. The proposed attempt is based on the algorithm developed and applied to calculate the real costs of energy under the given circumstances of demand and possibility of loading the renewable energy sources, a type of hydro, pumped storage and wind power plants – HPP, PSHP and WPP. The cost should be calculated for different types and sets of power plants and networks and finally should define the optimal policy for developing the generation structure for the given LEM.

CALCULATING THE POWER LOAD OF THE RENEWABLE SOURCES OF ENERGY

Electric power demanded by final energy consumers reflects the changes in the level of generation, transmission and distribution cost (Sroczan 1996 and 1999, Baltierra 1998). The cost of transmission and distribution is negligible in the discussed PS, with constant configuration of network.

The PS operator, realizing the EU directives, defines the assumed value of the load covered by ecological sources. In the presented attempt it is the RER – renewable energy ratio, defined as contribution of HPP and WPP to the PS generation process (Sroczan 2005).

The RER value is calculated as:

$$RER = \frac{A_{RE}}{A_T + A_{RE}} \qquad (1)$$

where: A_T – volume of energy generated by thermal units, A_{RE} – volume of energy generated by HPP and WPP.

The RER value is varied due to changes of γ and κ, coefficients converting the real value of energy generated by hydro and wind plants respectively. Therefore the developed simulator can support the decisions of PS manager in the area of power units loading priority.

The set of committing power units, operated in real time mode, affects the real costs of generation in accordance with power plants frame work stated by transaction made on the energy stock. In the case of loading renewable sources of energy – wind and hydro power plants (WPP and HPP), the operator should know the disposal power, resulting from the current and predicted (short term basis) state of weather– wind speed and stored water resources.

The Structure of Simulator

To minimize costs of generated energy simulation routines are developed to support the decision of choosing the optimal structure of the sources in the given PS and time range. The developed simulator should support the decision of choosing the proper policy of extending (or modifying) the different structures of power plants in the given PS. If the procedures of local optimization of committing plants are omitted, the costs of delivered energy will increase more than it is necessary from a theoretical point of view (Sroczan 2005).

Calculating problems are occurring in PS including the power plants with limited energy production – WPP, PSHP and HPP. The limitation constraints of energy generation refers especially to ecological energy, which additionally, except the PS and consumer effects, depends on wind and hydro resources. The costs of energy generation are calculated with regard to energetic characteristics of committing power units (Sroczan, Urbaniak 2001).

Calculating of energy cost

Wind farms consist of autonomous power plants generating electric energy in accordance with wind speed and rotor diameter:

$$P_{WPt} = \frac{\pi D_R^2}{4} \cdot c_p \cdot \rho_o \cdot v_t^3 \eta_r \cdot 10^{-3} \text{ [MW]} \qquad (2)$$

where: D_R- diameter of rotor of wind turbine [m], v_t – instantaneous wind speed [m/s], ρ_o – air mass [kg/m^3], c_p – wind turbine efficiency ratio [-], η_r – resultant coefficient depending on efficiency ratio of mechanical gear, generator and transmission network.

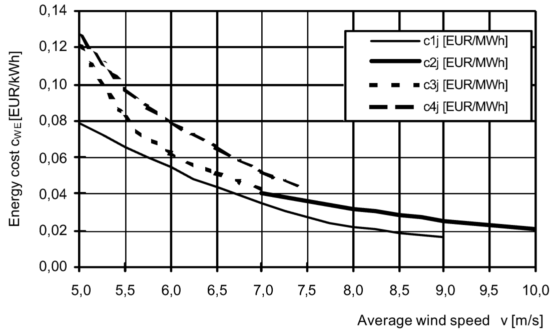


Fig. 1. The cost of wind-generated electricity depending on average wind speed and discounted cost of power plant.

Assuming the possible values of wind speed (annual average speed) in the given wind farms and rated power of units, it is possible to estimate the values of energy prices. For the wind generators the price of energy, which satisfied the producers, depends on T_r – annual time of load with the rated power P_{WPr} and economic constraints: discount rate, payback time, capital cost of investment and cost of maintenance and operation.

The calculated values are obtained using the methods of discounted cash flows in order to provide the answer to whether the investment is profitable or not. In particular the methods, total cost of ownership (TCO) or net present value (NPV) and internal rate of return (IRR), are applied. Usually the NPV is calculated as:

$$NPV = C_{Fo} + \sum_{t=1}^{n} \frac{\alpha \cdot C_{Ft}}{(1+r)^t} \qquad (3)$$

where: C_{Fo} – capital investments cost, C_{Ft} – cash flow in the tth-year, α- risk factor, r – discount rate, n – lifetime of power plant, t – year of exploitation.

For the purposes of simulation the values of renewable energy prices are calculated for typical PS with regard to wind speed. Because of the value of WPP power, defined in equation (1), it is highly sensitive to wind speed then to the others coefficients.

Example results are shown in Fig. 1. They illustrate the prices at the final consumption level, without the transmission and distribution cost. The values are obtained for the given lifetime of discussed wind farm and discount rate. The economy load of WPP is calculated using the methodology described in the paper (Sroczan 2005).

Power Balance

Power balance considers the relationships between: power plant, energy stock, provider, distributor, and consumer, in t-*th* hour of T – considered time period of calculation:

$$\sum_i P_{it} = P_{PSt} + \Delta P_{FLit} \pm \sum_k P_{bkt} + P_{rt} \qquad (4)$$

where: P_{PS} – demanded power of PS; ΔP_{FL} – power losses in given network branch; ΔP_b – stated range of balancing power, operated by the PS auto-frequency control system or PS operator, P_r – power system reserve, i – power plant number.

The volume of calculated wind-generated energy depends on T_{RP} – annual time of load the given WPP with its rated power P_{WPR}:

$$T_{RP} = \frac{\sum_{k=1}^{N} \eta_k \cdot P(v_{avk}) \cdot \Delta t_k}{P_{WPR}} \qquad (5)$$

where: $P(V_{avk})$ – average wind speed in k–th time period, Δt_k – time period defined for the given k–th range of wind speed.

Energy Generation Structure

The structure of the typical PS consists of some types of unit. Sources of energy defined as ecological (or renewable energy) are classified as wind and hydro power plants depending on weather constraints.

The final price of the electric energy for the delivery company is calculated as an average value, defined in the contract between power plant and wholesale energy provider for the given generated power and time range.

The results of the calculation of the energy cost in accordance with average wind speed (Fig. 2) and annual time of full load (with rated power) of the WPP are shown in Fig. 1.

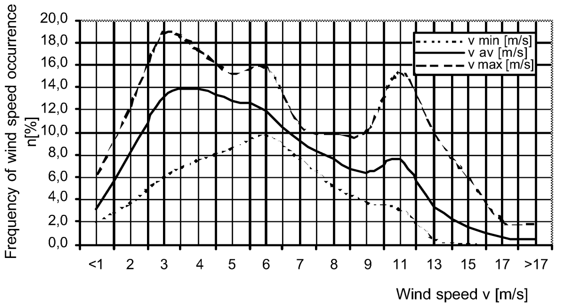


Fig. 2. Annual average and extreme values of wind speed occurrence measured for the given allocation.

The price of power and energy with regard to ecological sources depends on the PS manager's decision and LEM operator, if the decision is optimal the costs will fulfill, in each time *t,* the following relationship:

$$C_t \to \min\left\{\sum_{i=1}^{n} C_i\left(P_{gi}\right)\right\} \qquad (6)$$

where: P_{gi} – the level of generated power in i-th committing power plant.

RESULTS OF SIMULATION

The developed simulator runs in the interactive mode. The initial value of coefficients γ and κ is obtained upon the base of rated (or measured) energetic characteristic of thermal units. The value of RER is fixed as a result of PS managing policy and power units structure of PS.

The relationship between the increasing cost of generation by thermal plants and γ and κ coefficients (Sroczan 2005) affects the structure and the value of the load of the subset of HPP and WPP. The optimal case for renewable sources is obtained when $\gamma\frac{\partial W}{\partial W} = \kappa\frac{\partial C}{\partial P} = \min\left\{\frac{\partial C}{\partial P}\right\}$ and both types of renewable sources of energy – sets of HPP and WPP are loaded with rated power in the base of PS load. The stated volume of "green energy" is defined by PS operator, using the RER coefficient.

An increase in RER ratio affects the average cost of generated and consumed energy. In the example loaded set of units, when RER is increased in the range 50 % the decreasing of cost of generated energy approximate 12 %.

CONCLUSIONS

The effect of PS sources structure on generation costs is solved using some procedures of simulation. They enable calculation of the expected costs of energy for the given wholesale providers and power plants, assuming the varied structure of sources – types of power plant.

On a short time basis the results support the real time operation. Over a longer time period, they allow the definition of the optimal policy of extension or modernization of the power structure.

The structure of PS is important in case of failure or inconvenient weather conditions, for example too small or too great a wind velocity for WPP or weak rain and flows in the hydro plant. Optimal relationships among the stated power of all sources depend on both the power source set and consumers' behavior.

The developed simulator generates the range of acceptable values of λ, γ and κ coefficients balancing the load of committing units with regard to ecological sources of energy.

The defined RER – renewable energy ratio - allows the simple realization of generation policy in accordance with given contribution of renewable energy sources in the power plant set.

REFERENCES

Baltierra A.E.; Moitre D.; Hernandez J.L.; Aromataris L. 1998. "Simulation of an Optimal Economic Strategy of a Wholesale Competitive Electric Energy Market". In *Proc. of 10th European Simulation Symposium.* Nottingham, England 1998. 255-259.

Sroczan E. 2005. "The Simulation of Power System structure effects on technical and economic effectiveness of energy generation". In *Proceedings of the 2005 Simulation and Modelling Conference. University of Porto Portugal. 24-26 Oct. 2005.* 391-395.

Sroczan E. 1999. "Power System Energy Generation and Delivery Costs Simulation using Fuzzy Logic and Neural Networks". In *Simulation in Industry.* Horton G., Möller D., Rüde U.(Eds) 11th European Simulation Symposium Erlangen-Nurnberg 26-28 Oct. 1999. 399-403.

Sroczan E.M., Urbaniak A. 2001. "Simulation of routines of power system manager's decision effecting the natural environment". In *Simualtion in Industry.* Giambiasi N., Frydman C., (Eds.), SCS Europe BVBA Publ. Marseille, (France) 2001. 488-492.

Sroczan E. 1996. "Application of Artificial Intelligence to Algorithms of Power Plant Identification with the Purpose of Load Dispatch". In *Proc.of 8th European Simulation Symposium ESS '96 " Genoa.* Society for Computer Simulation International. Genoa. 1996. 592-596.

EUGENIUSZ SROCZAN is employed as an assistant professor at the Poznan University of Technology (PUT) and professor of State Technical High School in Gniezno. Obtained from PUT a M.Sc. and Engineering Degree in area of Industry Automatic and a Ph.D. in area of Electric Power System Engineering. Author and co-author of papers on Power System Economic Operation, Energy Management Systems in Industry and Automation of Energetic Processes as well as Water and Waste-Water Treatment Plant. Author of the book on contemporary electrical installations of home. Since 1984 is the President of Branch of Polish Electriciants Society at the PUT.

RESOURCE FLOW AND PLANNING MODELLING

The model of wood resource flow

Janis Oss

The Latvia University of Agriculture
Faculty of Forest

Asteru 10-53, Jelgava, Latvia, LV-3001
E-mail: janis.oss@e-koks.lv

Abstract

The authors have analyzed a wood resource flow. From the obtained data they have developed the model of resource flow. The main goal of the model of wood resource flow is to indicate the problems existing in the flow and to analyze future scenarios. The paper comprises description scenarios of the development of the wood resource model, results and simulation methods to check the function of the model.

Key words:

Forestry, Model design, Dynamic modeling, Corporate planning, Industrial controlo

Introduction

Latvia is rich in forests, because forests cover about 45% of its territory. In Europe forests cover on average 33% of the land.

In 2004 there were 10.75 million m^3 of wood harvested. Forestry is one of the most important sectors of the national economy of Latvia; its contribution for the GDP is 7.5%.

The transport sector is closely related with forestry, particularly - small ports and cargo transport, thus making it very sensitive to any changes in this sector.

The increasing use of wood resources raises the question - how appropriate it is used. Still the question exists, how appropriate is the processing of wood resources and how the waste wood is utilized.

By analyzing the flow of wood resources it is possible to obtain data about the types of end-products and waste wood. To obtain data concerning the present situation in the flow of wood resources, we must realize the process of flow from the place of the resource acquisition – the forest, to the processing (timber production, veneer industry, etc.).

In the flow of wood resources three sectors are involved: the sellers of wood resources; the transporters of wood resources and the customers of wood resources (pre-processing enterprises, ports). Each of the three sectors provides information. In analyzing this information we can determine the flow of resources and potential scenarios.

To perform the analysis of the wood resource flow it is necessary to develop a model, which describes the flow. There all sectors should be included into the model, which are connected with the resource flow. The common model of wood resources characterizes the market on the whole. In order to understand some of the processes deeper, there scenarios should be developed, which can be tested by means of simulation methods.

Materials and methods

The main aim is to develop the model of resource flow, which can characterize and analyze the future flow.

There are three sectors singled out of the model of wood resource flow, which can characterize the entire flow. These sectors are interdependent and when including them into the model, we should be aware exactly of the extent of their influence and the factors that could affect the flow.

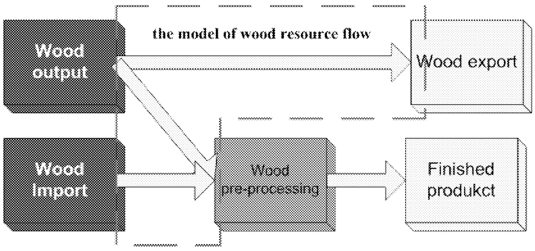


Fig. 1. The model of wood resource flow

In the beginning there is a part separated from the entire flow of resources, which characterizes the flow from the forest to the pre-processing object. The exactness of the model results, depends on the input data and the factors that regulate the flow.

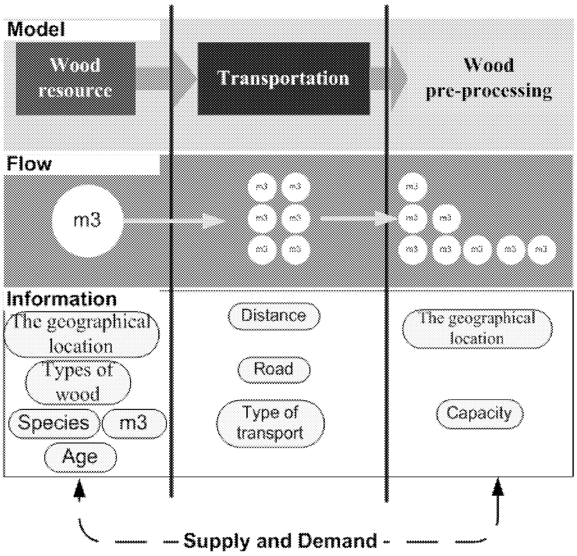


Fig 2. Model and type of flow

The geographical location of wood resources and pre-processing enterprise is very important for the functioning of the model, because one of the flow indicators is the distance between the location of wood resources and the pre-processing.

There the types of wood resource transportation should be included into the model. In Latvia the services of road and railway transport are used for transportation. The railway services are the most connected, with the import of wood resources, but for the inland services the railway, in fact, is not used.

Here is the scenario is developed, which shows the flow of wood resources from the geographical location of wood resources to the nearest pre-processing enterprise. When analyzing the data of this scenario, we can obtain information about the assortment demand. Using this scenario we can analyze the situation, when there is a pre-processing enterprise established near the wood resources, to answer the question: How long will the amount of wood resources be sufficient, if felling remains the same?

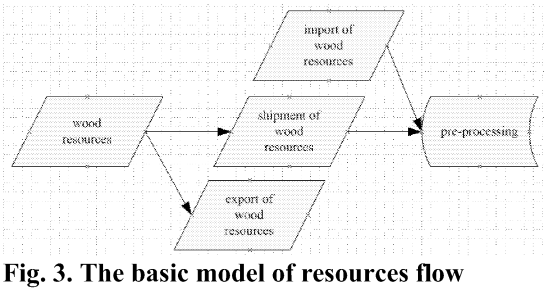


Fig. 3. The basic model of resources flow

The second scenario characterizes the demand of the wood resources. By indicating the pre-processing enterprises that process the same assortment we determine the factor, which influences the flow of each enterprise. Using this scenario, we have the possibility to evaluate, whether the enterprise would have sufficient amount of wood resources and how much other enterprises can affect the flow.

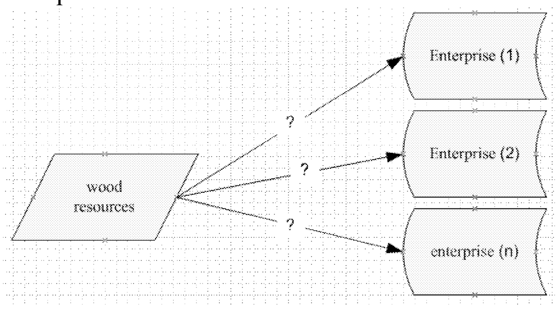


Fig. 4. Scenario of demand

Results

The developed system of wood resource flow comprises a set of programs, which realize the data processing and their testing, using the system simulation.

There has been a system developed, which includes the geographical information system "ArcGIS" with "Network Analyst" extension and the modeling/simulation software "EXTEND" Industry.

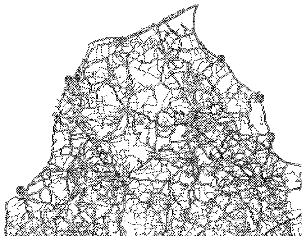


Fig. 5. The network of roads in Latvia.

The task of "ArcGIS" is to process and pass along the data of the distances, road surfacing and the speed of driving between the geographical location of wood resources and pre-processing enterprise.

There has been a network of the roads in Latvia developed, whereupon the layer of the location of enterprises and wood resources are included (see Fig. 5). Using the functions of "Network Analyst" the advantageous roads and distances for driving are analyzed. Most of the wood-roads are influenced seasonally. This influence indicates, whether in a particular season the road can be used and what restrictions should be observed concerning the speed and the size of consignment. Here are the data is passed along to the data processing, which determines the performance of model scenarios.

The main process, where the data are analyzed and the simulation of flow is carried out, is developed using "EXTEND" Industry. There are standard components used, as well as special ones developed, in order to work out the model. The positive feature of "EXTEND" is that it is possible to form ones own components. ModL programming language is used within "EXTEND" system; the syntax of this language is similar to that of C++ language.

"Discrete Event" model is developed on the basis of Extend methodology. In "Discrete Event" model it is possible to add additional items to each value. The flow is formed using the data of items.

Modeling of "Discrete Event" is chosen on the basis that the addition of data and their size are not cyclic. This positive quality enables to input the values of any size into the model at any time.

The following division of a value and positions is developed in the model of wood resource flow:

- Value – the quantity, which should be included into the model. Each unit is equal to 1 m^3.

- Owner, wood species, assortment, wood age - items, which are added to the value as additional information.

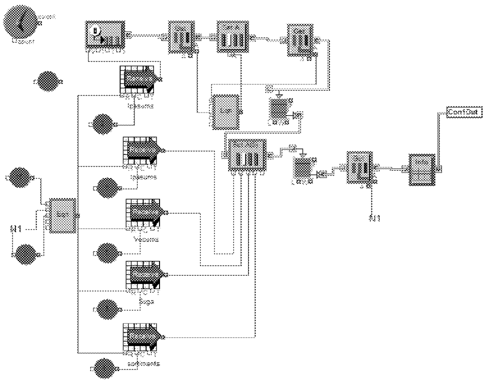


Fig. 6. The input into the model and the defining of items.

During the functioning of the model additional items are added to the unit, which are used for the analysis of data and the control of flow. At any stages of the model functioning there is a possibility to obtain the information on the unit that flows through the blocks.

The main parameter of the model functioning is the wood resources. At the initial stage of the model the amount of resources is provided in cubic meters, and to this quantity there are items added that identify the data (owner, wood species, assortment) – see Fig. 6.

An original model is developed according to the principle – the resources that will be received by the customer, whose ratio of distance and purchasing price is the most advantageous. When finding such a customer, the testing is carried out concerning the correspondence of the ordered assortment to the supply. If this criterion does not correspond, the next most advantageous supply is accepted.

The data flow through the blocks as a separate unit is equal to $1m^3$. This feature enables to control each of unit separately and to direct the flow more exactly. Of course, there is the possibility to combine each unit in a larger package. It is used during the transportation of resources – the size of a package is determined according to the transport to be used. The size of the package can differ not only by the type of transport, but also in compliance with the law, which provides the load of cargo in a particular season.

The last part of the wood resource flow is a pre-processing enterprise. In order to let the enterprise be involved in the model, it should have certain factors that characterize it. The model provides that the enterprise is characterized by the following parameters: the geographical location of enterprise, assortment and its amount, which enterprise would like to purchase, and the processing capacity of the enterprise.

At present the situation of the parameters, characterizing the flow, are kept in the db table. As it was mentioned above, there should be access to the parameters for each object, involved in the model. The functioning of the model depends on these parameters.

Conclusions

1. The developed model of wood resource flow can be used to analyze the situation in the market, the competition, and the factors that influence resource flow.
2. By combining the model of wood resource flow and the information database on the wood resources it is possible to perform the analysis of a real market situation, as well as the future development alternatives.
3. In order for the model to function perfectly, it is necessary to develop a system by means of which the data are imported into the flow model automatically.
4. To develop the functioning of the model, it is necessary to work out a system that would ensure the data exchange with the SQL database.

Bibliography

1. (2002.), Extend v6. User Guide, United States of America: Imagine That.
2. John J.Coyle, Edward J.Bordi, C.John Langley Jr (2003), The Management of Business Logistics: A Supply Chain Perspective, 7e: Ohio, Canada, Transcontinental Lousiville, Quebec
3. Heikki Juslin, Eric Hansen (2003), Strategic Marketing in the Global Forest Industries: United States of America, EP Imaging Concepts, Shelton-Turnbull Solutions
4. Ronald H.Ballou, (2003) Business Logistics/Supplu Chain Management: , Prentice Hall College Div.

Event simulation of supply chain networks – Dynamic detailing in the material flow simulator d^{3}FACT insight

Wilhelm Dangelmaier
Mark Aufenanger
Kiran Mahajan
Chrtistoph Laroque
Daniel Huber

University of Paderborn
Heinz Nixdorf Institut
Fürstenallee 11, 33102 Paderborn, Germany
E-mail: {whd|marka|laro}@hni.uni-paderborn.de

KEYWORDS
Digital factory, event simulation, material flow simulation and visualization, d^{3}FACT insight, supply chain network, production networks

ABSTRACT

Customized production and development, growing cost pressure during the production and shortened product life cycles direct to coalescence of a virtual factory with all partners involved within the manufacturing process of a product. Replaning and rescheduling of a partner inside the supply chain can, because of dynamic effects, direct to other effects, which up to now could not be foreseen and not assured or only with difficulties within the planning. The complexity of the supply chain prevents until now the simulation and visualization of these effects because it could no longer been scheduled in real-time. This article shows a possible mapping of a production network within the material flow simulator d^{3}FACT insight, which allows such a simulation regarding the method of dynamic detailing for the calculation of simulation models during their execution. Every participant of the supply chain was able to identify possible sources of defect within an integrated simulation model before real implementation and to adjust and organize the implementation. Therefore whole activity of the supply chain could be designed more successfully.

MOTIVATION

As a consequence of the shortened product development time and product life cycles, an increasing number of variants and the from there raising requirements concerning a higher flexibility of production- and purchasing processes, the collaboration of producing organizations and their suppliers within supply chain networks is growing. Already today, rough demand numbers of the OEMs in early phases are transmitted to the victuals and be verbalized precisely over more iteration steps. The whole supply chain up to the delivery of the finished product through the OEM to the customer sees itself as a single, virtual company. Very often in the practice there is a phenomenon to find, as the replaning and rescheduling within a process step by a respective participant is taking place without a feedback between the previous and subsequent process steps. The influences of the dynamic effects inside of the supply chain are neglected and became visible not until the realization.

The adoption of the material flow simulation offers itself as an instrument of identification and visualization of these effects. Assumption is a precise mapping of every process step of the whole supply chain within an integrated simulation model for the event simulation. This mapping of all production steps inside one integrated simulation model, normally guides to a very complex simulation model, so that a simulation and concurrent real-time animation on one normal computer is no longer realizable. The simulation and visualization of the occurring dynamic effect was forbidden by itself until now. By the use of the method for dynamic detailing of simulation models during their execution, the wanted simulation of the whole supply chain of a production network becomes possible. Therefore the calculation of the simulation model during its execution becomes reduced to a maximum, still in real-time calculated admeasurements, which could be animated by the visualization component.

Thereby the real simulation model adopts the level of detail which should be simulated by using different criteria during the runtime and simulates relevant process nodes by using a high level of detail. Process steps which are not significant for the accomplished changes are not simulated in detail, wherewith additional computing time could be saved. The event simulation of the whole supply chain could become possible. If furthermore the necessary security aspects of every partner within the supply chain by the use of a rights management inside the simulation environment are guaranteed, the realization of the simulation of a complete supply chain becomes realistic.

STATE OF THE ART

For the virtual planning and safeguard of production processes, the material flow simulation is an established method since years (Law and Kelton 2000). Software tools like UGS eM-Plant, Delmia’s Quest or Taylor ED by Enterprise Dynamics (Mueck and Dittmann 2003) are used for material flow simulation regularly (Bayer et al. 2003). With these tools it's possible to create, to validate, to verify and to compute models of the focused production process.

In all these simulation environments the simulated model is regarded as a static scene during the simulation experiment. The modeled scenarios, even those, which are not adapted to real world, stay constant over the simulated time. Especially in the area of ramp-up-processes, where the layout and processing order of the simulated production changes over the time or new variants are integrated in the simulated process, This behavior leads to significant problems in modeling a simulation as well as the communication of the simulation results.

Today's simulation projects are moreover so substantial and complex, that they are belabored in simulation teams. Beside project members from different areas of application, normally several simulation experts are working on one simulation model. Nowadays they are nearly unsupported in their collaborative work.

The user himself regarding a simulation run is traditionally just a passive viewer of a computer-generated (mostly 2-dimensional) scene, without having the possibility to interact in any way with the actual calculated simulation experiment. The theoretical existing potentials of a three-dimensional visualization, as it is supported in some simulation tools, are not made accessible. A more realistic user integration in the calculated, virtual environment can improve his understanding of the process by the higher immersion; it's possible to create a more meaningful planning environment. A combination of layout and process planning in a three-dimensional modeling environment enables the user to recognize additional restrictions during the creation of the simulation model from the beginning. The quality of the result can be improved and additional planning time can be saved.

d^{3}FACT insight

As simulation environment for the implenetation of supply chain models and methods for dynamic detailing, the material flow simulator d^{3}FACT insight (Dangelmaier et al. 2005), which was developed at the Heinz Nixdorf Institut, is used. The tool is a event based modeling and simulation environment comprises of different modules (Figure 2).

Beside the support of a concurrent team work of multiple simulation experts on one simulation model, especially the improvement of modeling and simulation by the execution in a three-dimensional environment was one goal of the development of d^{3}FACT insight. The user itself is integrated in the highest possible way in the simulation scene. Moreover, new procedures within the project have lead to advances in the daily modeling work. By a special method, it's possible to adjust the resolution of a material flow model dynamically during its execution.

Starting with a graphical user interface, the simulation model is created in d^{3}FACT insight by the user. It's defined in a flexible and extendable data format (XML) and saved in the simulation database. During the saving process, the simulation model is compared concerning its structure to the existing description (DTD). If the simulation model is to be executed in the kernel module of the tool, it's translated from the XML-format to a compiled Java™ program (the existing file is transformed to Java™ code by the use of XSL and compiled by a preprocessor). The resulting archive (jar-file) can be shown and executed in the different visualization modules. Figure 1 draws a picture of the transformation process.

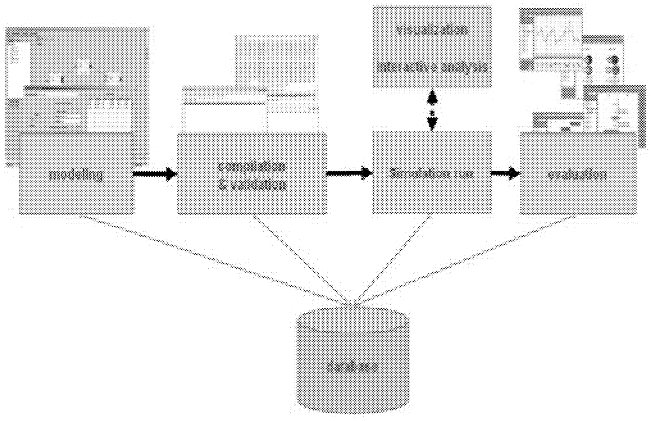


Figure 1: Process of model transformation

Before the intrinsic start of the calculation of the simulation experiment, the initialization is executed, where the simulation is filled with the input data from the simulation database and starts the calculation. An experiment manager is to manage several simulation experiments, so that some simulation experiments can be computed sequential or parallel on different computers. Subscribed variables are recorded in the building blocks of the simulation model or are stamped on the tokens, which run through the system. This collected data is saved in the database and is analyzed sub sequentially. Because of the possibility of interaction for the user, during the execution of the simulation model, the parameterizations made by the user are recorded as well. During the experiment, the variables marked as vr_visible, can be viewed and in some cases changed. The analysis of the simulation experiment can be adjusted individually. Some standard analysis and statistics are presented by standard building blocks, available in a modeling library.

The object-orientated programming language Java™ is a basis for the development of the simulation tool. Java™ is widely used in most IT-areas in industry, commerce and administration and is available for most operation systems and platforms, from mobile phones up to real-time mainframes. There are lots of libraries available, which can be used in any way in the simulation tool. Its attributes as an object-oriented language allows the use of inheritance, hierarchies and data casing as it's required in simulation models and their building blocks. JAVA is easy to learn for the system user, because there are lots of tutorials, books and materials available. An integration of the tool into the running processes of a company is thereby lightened enormously.

The separation in modules follows functional aspects, and was made as described by Figure 2.

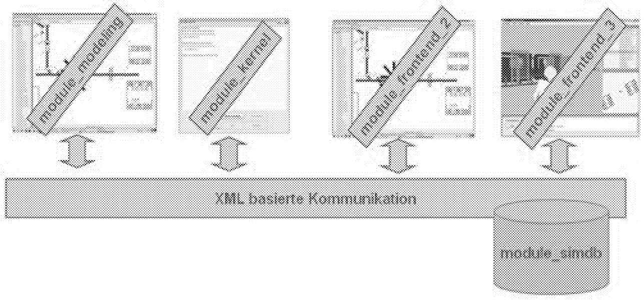


Figure 2: Modules of d³FACT insight

The 2D, 2.5D or 3D-dimensional modeling component allows the fast creation of complex and hierarchical simulation models. Building blocks can be arranged in libraries and for every building block; it's possible to connect a three-dimensional representative from the simulation database, so that the generated scene can be displayed in the 3D-frontend (Mahajan et al. 2005). From the 2D-adjustment of the building blocks, the 3D as well as a 2D or 2.5D-visualization can be derived. The 3D visualization front-end is developed with new rendering methods, so that even large and complex scenes can be displayed during a real-time simulation run (Fischer et al. 2001) (Klein et al. 2002).

The selected simulation model, started from one of the visualization environments, is to be computed in the simulation kernel, which works like a discrete, event-driven manner. Besides the internal simulation time, the simulation kernel synchronizes the real-time of the different clients.

An efficient analysis of the modeled material-flow, for example at the beginning of a simulation experiment, can be solved by the use of this visualization module. For the developer of a simulation models, it's possible in a very early state of the simulation project to detect bottlenecks and a general view on the system's behavior in a simple way. This environment can also be used for simulation projects, which focus more likely on numerical analysis than a realistic 3D-view on the modeled scene. Thus it's also possible in the two-dimensional visualization to interact with the simulation model during the execution of a simulation experiment, in order to make the effects taken on the model visible.

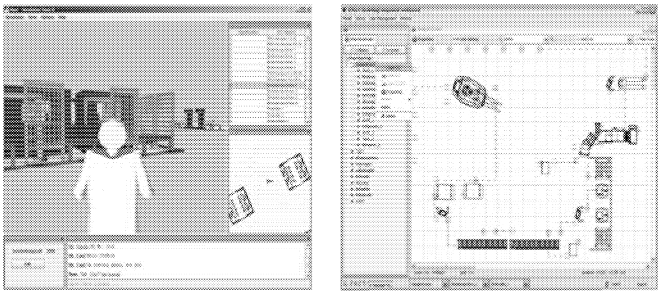


Figure 3: Visualization of a scene in 2D/3D

The following section describes the implementation of a method, which allows an online-adoption of simulation models in d³FACT insight.

METHOD FOR DYNAMIC DETAILING

The implemented solution uses the method of the inhomogeneous dynamic level of detail (Dangelmaier and Mueck 2004), whereby the simulation models are available in different levels of detail. By the use of a dynamic adjustment of the simulation model during the runtime, dependent of the users' position or the behavior inside the virtual scene and significant process points, it is possible to ensure the highest possible detailed simulation in the required accurateness.
The following graduations are intended in this method:

- High level of detail in all areas, the user is putting his attention on
- High level of detail at process point, where significant processes happen
- Middle level of detail in all areas, where the user could put his attention in the near future
- Low level of detail in all the other areas

A simulation models level of detail becomes dynamically calculated by the users' position and his line of sight in the virtual scene. The users' behavior is evaluated by different indicators and depending on the result, an adaptation of the scenario was accomplished. These adaptations are roled back into the simulator where they are used for the calculation of the simulation models level of detail. The level of detail is independent form the visualization techniques used for the real-time rendering of the 3D scene, it exclusively relates to the level of detail of the actually simulation model. These methods allow a dynamic calculation of the areas which are under users' attention, by the interaction of the user and the virtual environment. By the use of this information the level of detail for the corresponding areas is derived. Additionally to the calculation of the indicators, the switching processes have to be started and calculated. They are switching the elements of the simulation to the calculated level of detail.

User stimulated switch criteria

For the beginning there are three methods for calculating the switching point realized which can be used individually or as combination. Besides the users position within the model his line of sight is a criteria for the calculation. The third criteria based is based on the linking of the modeled building blocks inside the simulation model. Another criteria relates to the objects inside the virtual scene, which are masked by other objects and therefore are not visible for the user. These criteria could only be calculated by the visualization module and therefore it is inapplicable for a calculation by the simulation kernel.

Users' position

A human recognizes the more far objects smaller and in a lower quality as objects which are near to his position. This is applicable for three dimensional, virtual environments too.

The more far away objects are getting a smaller watching focus. Therefore they can be visualized in the lowest level of detail.

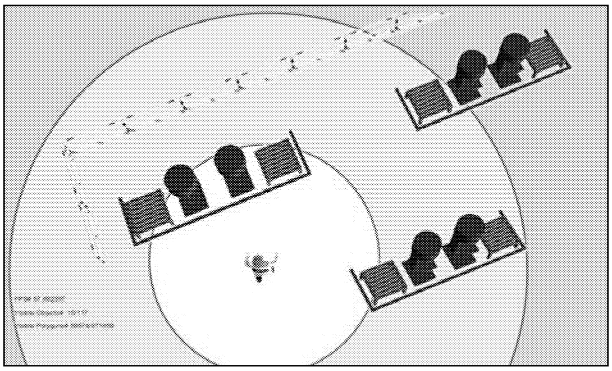


Figure 4: Indication by distance

The distance between the user and a simulation object was calculated by his position and the position of the nearest point of the objects surface.

The number of the different levels of detail is variable; any number distances could be set as boundary values for the level of detail. Simplifying they become displayed as circles, within its center the user has his position (compare Figure 4).If there is an object within the inner circle or is cutting the circle, by the use of this indication method, it should be simulated in the highest level of detail. Due to the the probability that a users put his focus on objects close to him, the quality of this method could be appraised as high. Furthermore indication by the distance could be calculated very fast.

MODELLING AND SIMULATION OF SUPPLY CHAIN NETWORKS

The execution of the simulation of a complete supply chain needs the single process steps (distributor and OEM) as a multilevel simulation model with different levels of detail. A single distributor has to be described by different detailed models. Figure 5 shows a cutout of a possible production network. *Distributor A* is described by 2 simulation models with different levels of detail (black-box and level 1). The number of levels of detail in principle is unlimited.

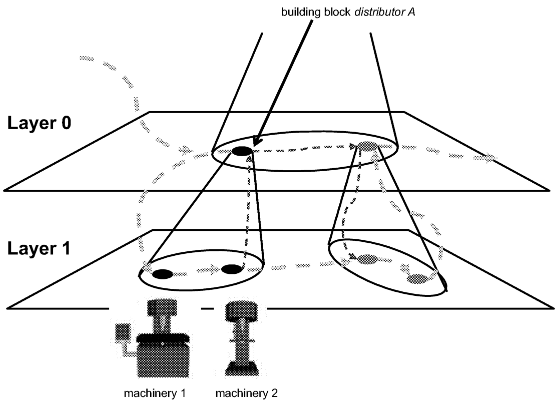


Figure 5: Simulation model of the supply chain with different levels of detail (cutout)

During the execution of the simulation run, at first all the model parts of the executing process step (for example of a distributor within the production network) are simulated at a high level of detail. The assumption is, that the own simulation models are relevant. The other process points, other partners within the integrated supply chain are simulated on a lower level of detail. If a change which was made leads to a significant effect in the other process point, they become simulated by the use of the stimulation of the marginal values. Therewith the changes made by the user became presented in best possible way. The working load of the executing computer describes the upper bound, that is the fact if the computer has to be working to full capacity there are no further switching in higher levels of detail possible, to ensure that the execution in real-time could be still done.

The multi-user functionality and its assigned rights management can control the access to potential secret functions within the simulation model that is in face of a collective execution of the whole model, a manipulation and adaptation of the model can be bounded.

CONCLUSION

By use of the implementations presented in this paper, different opportunities of simulating dynamic switching operations in detail variant material flow simulation models are analyzed and realized. For this reason it is possible to calculate the watching focus of all relevant simulation objects automatically by the user interactions within the virtual scene and to put the objects into the appropriate level of detail.

Through this method it became possible, to simulate and visualize supply chain networks of producing companies in real time and to ensure the specific security requirements of the companies. Therewith the consequences of replaning at single process steps can be made visible and became recognized before the real implementation. Through the intensive adjustment of all the involved partners on the base of the integrated simulation model, all changes within the supply chain can be coordinated from the first and thus absorb the modification effects.

By the use of detail variant simulation an additional expense during the modeling phase is necessary, because the different model hierarchies of a system has to be represented. The research focus in this area currently lies on automatization approaches, which should be able to generate less detailed models by the use of the high detailed model in an (semi)automatic way. Thus the simulation model builder could be discharged in his work of building simulation models.

REFERENCES

Bayer, Johann, Collisi, Thomas und Wenzel, Siegrid (Hrsg.): Simulation in der Automobilproduktion, Springer-Verlag, Berlin u.a., 2003

Dangelmaier, Wilhelm; Huber, Daniel; Laroque, Christoph; Mueck, Bengt: d³FACT insight - An immersive material flow simulator with multi-user support. In: Bruzzone, Agostino; Williams, Edward (Hrsg.): *Proceedings of the 2005 Summer Computer Simulation Conference Bd. 37 SCS*, S. 239-242, 2005

Dangelmaier, Wilhelm; Mueck, Bengt: Using Dynamic Multiresolution Modeling to Analyze large Material Flow Systems. In: Ingalls, Ricki G.; Rossetti, Manuel D.; Peters, Brett A.; Smith, Jeffrey S. (Hrsg.): *Proceedings of the 2004 Winter Simulation Conference (WSC'04)*, S. 1720 - 1727, 2004

Fischer, Matthias; Meyer auf der Heide, Friedhelm; Peter, Ingmaer; Straßer, Wolfgang; Wand, Michale: The Randomized z-Buffer Algorithm: Interactive Rendering of Highly Complex Scenes. In: *Computer Graphics (SIGGRAPH 01 Conference Proceedings)*, S. 361 - 370, 2001

Klein, Jan; Krokowski, Jens; Fischer, Matthias; Wand, Michael; Wanka, Rolf; Meyer auf der Heide, Friedhelm: The Randomized Sample Tree: A Data Structure for Interactive Walkthroughs in Externally Stored Virtual Environments. In: *Symposium on Virtual Reality Software and Technology (VRST '2002)*, S. 137 - 146, Hong Kong, China, 2002

Law, Averill M.; Kelton, W. David: Simulation Modeling and Analysis, McGraw-Hill, 2000

Mahajan, Kiran; Laroque, Christoph; Dangelmaier, Wilheml; Soltenborn, Christian; Kortenjan, Michale; Kuntze, Daniel: d³FACT insight: A motion planning algorithm for material flow simulations in virtual environments. In: Schulze, Thomas; Horton, Graham; Preim, Bernhard; Schlechtweg, Stefan (Hrsg.): *Simulation and Visualization 2005 (SimViS) Bd. 1*, European Publishing House, S. 115-126, 2005

Mueck, Bengt and Dittmann, Nico. 2003. “Marktanalyse: Materialfluss Simulatoren.“ ALB-HNI-Verlagsschriftenreihe, Nr. 11

BIOGRAPHY

Wilhelm Dangelmaier was director and head of the Department for Cooperate Planning and Control at the Fraunhofer-Institute for Manufacturing. In 1990 he became Professor for Facility Planning and Production Scheduling at the University of Stuttgart. In 1991, Dr. Dangelmaier became Professor for Business Computing at the HEINZ NIXDORF INSTITUTE; University of Paderborn, Germany. In 1996, Prof. Dangelmaier founded the Fraunhofer-Anwendungszentrum für Logistikorientierte Betriebswirtschaft.

Mark Aufenanger studied business computing at the University of Paderborn, Germany. Since 2005 he is research assistant at the group of Prof. Dangelmaier, Business Computing, esp. CIM at the HEINZ NIXDORF INSTTUTE. He is mainly interested in material flow simulation.

Kiran R. Mahajan obtained his master’s degree in mechanical engineering with a specialization in production engineering from the Delft University of Technology in the Netherlands. He is currently working as a research assistant to Prof. Dangelmaier at the Heinz Nixdorf Institute, Germany since 2004. His research interests are development of simulation based planning and scheduling systems for complex manufacturing applications.

Christoph Laroque studied business computing at the University of Paderborn, Germany. Since 2003 he is PHD student in the graduate school of dynamic intelligent systems and research assistant at the group of Prof. Dangelmaier, Business Computing, esp. CIM at the HEINZ NIXDORF INSTTUTE. He is mainly interested in material flow simulation and the development of the “digital factory”.

Daniel Huber studied industrial engineering at the University of Paderborn, Germany. Since 2005 he is a research assistant at the group of Prof. Dangelmaier, Business Computing, esp. CIM at the HEINZ NIXDORF INSTITUTE. His main interests are material flow simulation, modeling methodology and automatic model abstraction.

MODELLING METHODOLOGY AND SIMULATION OF A HOSPITAL LAUNDRY

Michel Gourgand
Fateh Mebrek
Alain Tanguy

LIMOS, CNRS UMR 6158, Université Blaise Pascal,
Complexe des Cézeaux, 63177 Aubière Cedex, France
E-Mail: gourgand@isima.fr, mebrek@isima.fr, tanguy@isima.fr

KEYWORDS
Management of the laundry, Modelling Methodology ASDI, Discrete Event Simulation, SIMULA, WITNESS.

ABSTRACT
The laundry delivers clean linen and washes dirty linen of the care units. For well managing this service, the hospital Hôtel Dieu of Clermont-Ferrand sets up its central laundry at the new hospital Estaing. Various parameters influence the behaviour of this service such as the dimension and allocation of the critical resources. We propose an adaptation of the modelling methodology ASDI (Analysis, Specification, Design and Implementation) and simulation models to test functioning policies.

INTRODUCTION

Many researchers were interested at the design of information system for modelling hospital systems. Among them: (Benanteur et al. 2000) studied hospital logistics; (Combes 1994) worked on the load of the operating theatre suites and on the personnel planning. Recently new works were carried out on the new hospital Estaing (NHE), the main objective was to model and to simulate the hospital pool. (Andre 2005) studied modelling and simulation of the NHE logistics flows; (Gourgand et al. 2005) carried out simulation models of the brancardage concerning the transportation of patients; (Chauvet et al. 2005) studied the management of the NHE and designed, with ARIS, a knowledge model of the paediatric pool; (Mebrek et al. 2006) have modelled and simulated the imagery pool of the NHE. This paper introduces the laundry and the transit zone of the new hospital Estaing and then describes the modelling methodology adapted to its structure (knowledge model). Finally action models are presented and compared: a queueing network model is given as a basis for simulation models using SIMULA and WITNESS.

MANAGEMENT OF THE LAUNDRY

The laundry is an important logistic service of the hospital. It insures the distribution of the clean linen and the collection of the dirty linen in the care units of the hospital (figure 1). Currently, the nurses of the hospital occupy themselves of the management of the linen room. This situation generates an important wasted time within their care tasks. In the new organization, logistic agents replace the nurse for the linen task as well as the collecting of dirty linen, and the distribution of clean linen in the care units of the NHE. The transit zone is a small logistic platform which has three functions: transport, distribution and storage with inter-modal properties. It is also concerned by the transportation of meal, drugs and other products.

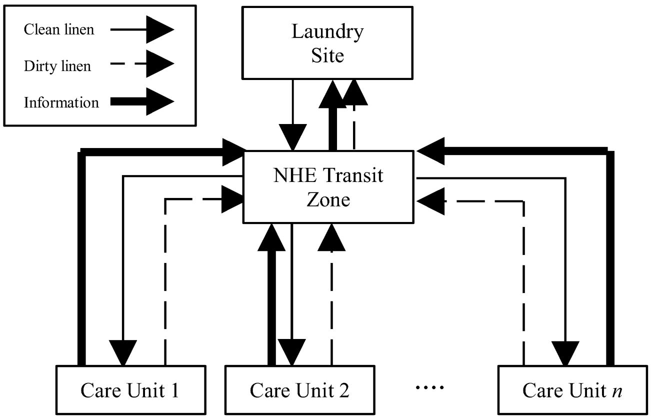


Figure 1: Linen flows

MODELLING METHODOLOGY ASDI

In this section, we introduce two concepts: the modelling process and modelling environment.

Modelling Process

The iterative modelling process (Figure 2) was first introduced in (Gourgand and Kellert 1991). It is composed of four steps:
- construction of a knowledge model based on the analysis and the specification of the system,

- development of an action model, using the knowledge model,
- exploitation of an action model,
- modifications or actions on the system.

The construction of the knowledge model must be carried out in collaboration with the experts of the system domain. This model must remain coherent in time, whatever its level of smoothness and the evolutions brought to the system at the time of its use. The objectives must be clearly identified in order to know the problems to solve. The construction of the knowledge model consists in collecting and formalizing the knowledge on the studied system. A functional and structural analysis formalizes the system in a written form. The specification must enable the modelling experts and the system experts to agree about the operations of the system.

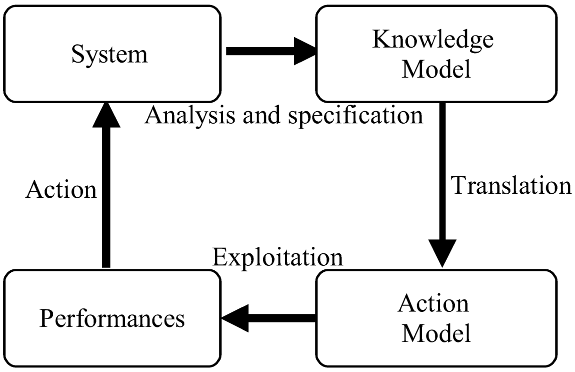


Figure 2: Modelling Process

The modelling methodology is mainly based on the definition of the knowledge model specification composed of the description of three subsystems and there interactions:
- the physical subsystem (PSS) consists in the physical entities providing a service or participating in an operation, the PSS topology defines the position of physical devices and their connections,
- the logical subsystem (LSS) contains the transactions treated by the system and the definition of services and elementary operations in regard to the treatment of transaction flows and depending on the entities in the system,
- the decisional subsystem (DSS) contains the management policies, resource allocation and system functioning rules.

Modelling environment

The main aim of the modelling methodology consists in building a knowledge model as generic as possible that enables the implementation of action models for specific systems of the domain. The knowledge model remains an open model that is enhanced by each study of hospital systems. The knowledge management and the implementation of action models implies the computer aid provided by a modelling environment that could be open in order to include new and accurate methods or tools.

The modelling environment (Figure 3) helps the user to exchange information with different partners of the project and facilitates the design and the implementation of action models during the phase of information extraction from the knowledge model. It is an attempt to the automation of the modelling process using knowledge formalization, data analysis, characteristic computation, operational research, evaluation, graphic and animation tools.

The first knowledge model of the hospital logistic system operations and structure is formalized by means of the software tool ARIS (Architecture of Integrated Information Systems), suggested by Scheer (Scheer 2002). This tool is suitable to describe organizations, processes and activities (Green et al. 2000), as well as entity relationship models (Chen 1976). Some parts of the hospital system are specified with the UML language. An extra simulation module is available for ARIS, but for graphical needs, financial, accuracy and policy reasons WITNESS simulation tool was preferred so as to design and to implement action models. Let us notice that the information extraction from ARIS files is not so easy that we may expect, and let us remark that WITNESS was specifically designed for industrial systems.

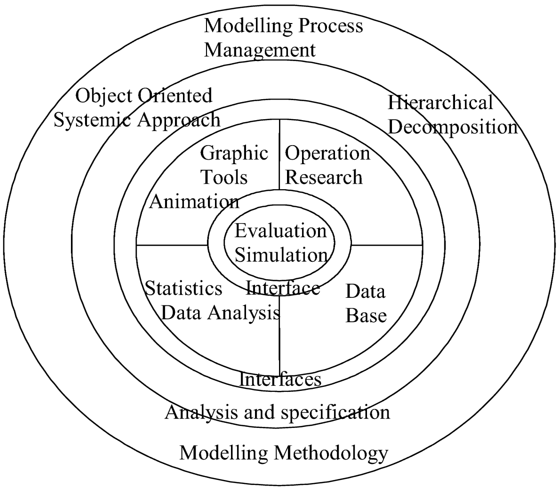


Figure 3: Modelling Environment

KNOWLEDGE MODEL

We built the knowledge model of the laundry service of the new hospital Estaing of Clermont-Ferrand using ARIS tool (Architecture of the integrated Information Systems) (Scheer 2002) in order to model it in the objective is to simulate it by using the software of simulation SIMULA and WITNESS in order to build a graphic model. ARIS is a modelling tool which is based on the processes.

The laundry process composed of two processes: the clean linen and the dirty linen. Both processes are described by event-driven process chains presented in figures 4 and 5.

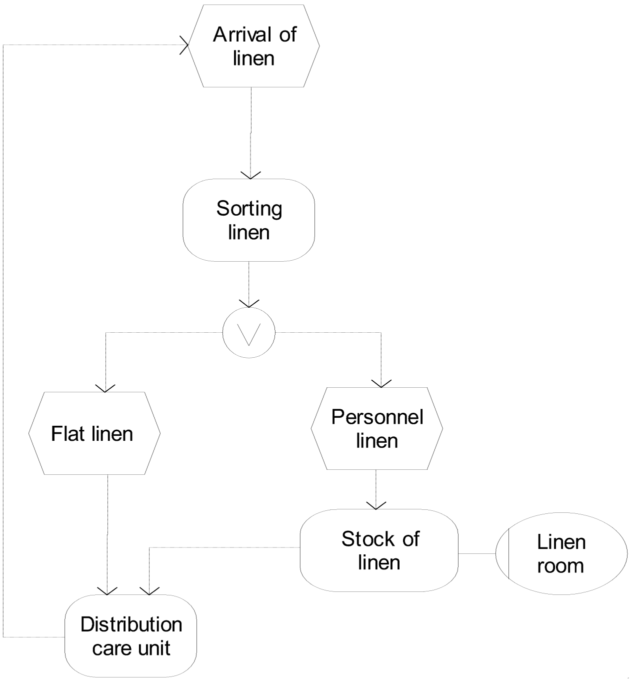


Figure 4: ECP of clean linen

Figure 4 shows the diagram of event-driven process chain (ECP) of the clean linen and (figure 5) dirty linen. The CPE enables us to give the detailed operation of a process of its beginning until the end of process as shows it above the diagrams.

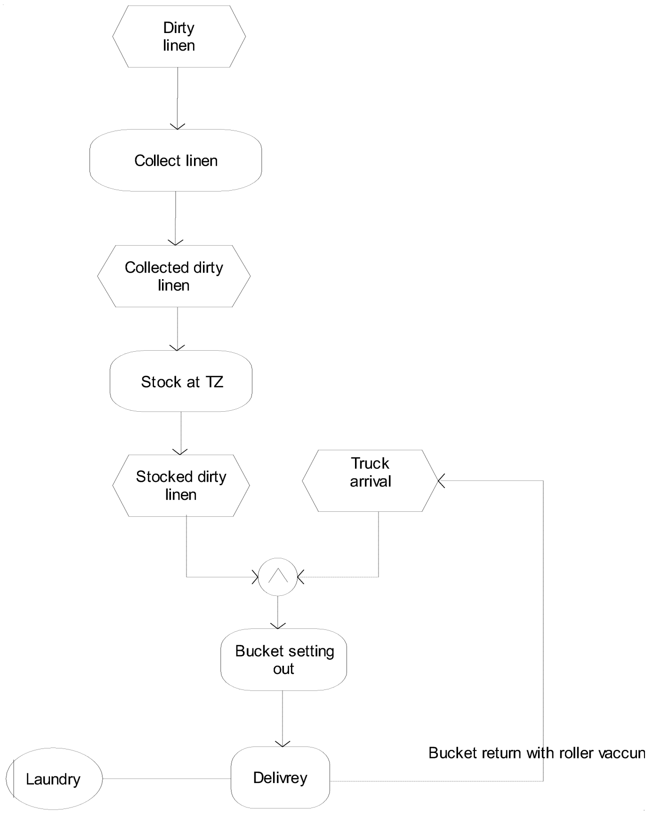


Figure 5: ECP of dirty linen

ACTION MODELS

The realized action models are simulation models based on the following queueing network model.

Queueing Network Model

Figure 6 describes a simplified queueing model. The total mission duration includes the following times:

- Loading of linen bags into carriage,
- Full displacement (clean linen),
- Unloading of carriage of linen bags,
- Loading of dirty linen bags,
- Empty displacement (dirty linen),
- Unloading of dirty linen bags.

This model is very simple but enable to estimate the resource utilization rates.

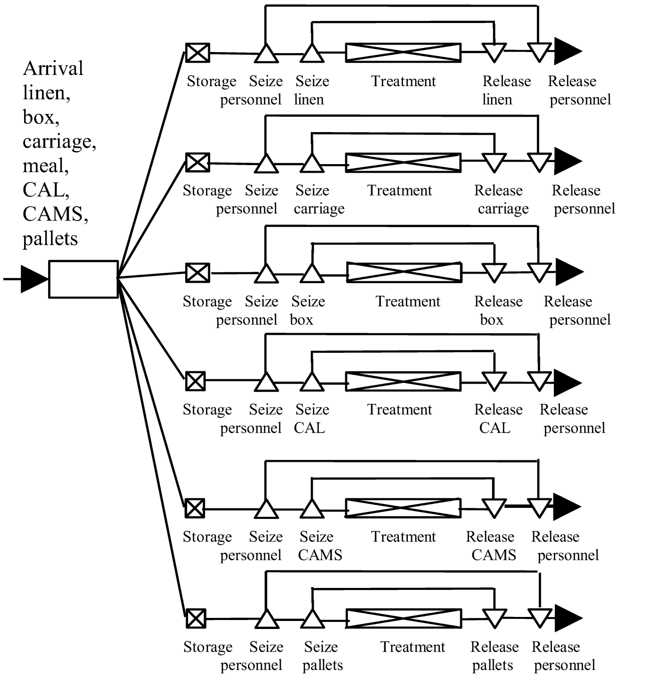

Arrival linen, box, carriage, meal, CAL, CAMS, pallets
Storage Seize personnel Seize linen Treatment Release linen Release personnel
Storage Seize personnel Seize carriage Treatment Release carriage Release personnel
Storage Seize personnel Seize box Treatment Release box Release personnel
Storage Seize personnel Seize CAL Treatment Release CAL Release personnel
Storage Seize personnel Seize CAMS Treatment Release CAMS Release personnel
Storage Seize personnel Seize pallets Treatment Release pallets Release personnel

Figure 6: Queueing network model

The duration of treatments is given by formula 1.

Treatment = Loading + Full displacement (clean linen) + Unloading + Loading (dirty linen) + Empty displacement (dirty linen) + Unloading (1)

The initial model uses uniform distributions to obtain an evaluation of various policies (table 1).

Table 1: Distribution of treatment durations

Treatment	Duration
Loading	Uniform(5; 10)
Full displacement	Uniform(10; 20)
Unloading	Uniform(5; 10)
Loading dirty linen	Uniform(1; 5)
Empty displacement	Uniform(3; 8)
Unloading dirty linen	Uniform(5; 10)

Table 2 contains the linen quantities for each care unit (CU) of the NHE.

Table 2: Distribution of quantities (bags)

Care unit	Quantity
CU1	3
CU2	20
CU3	5
CU4	18
CU5	9
CU6	12
CU7	21
CU8	7
CU9	4
CU10	15
CU11	13
CU12	19
CU13	20

The main objective of this study is to reasonably increase the utilization rate of the resources and to decrease the latencies in the service of the laundry. A lot of data are collected but many important ones lack, and we need validated estimations for the NHE. The probability distributions are not exactly known. The arrival processes are not Poisson's processes. We first use uniform distributions because their parameters are easy to obtain, they never give unwanted negative durations (a Gaussian distribution may provide such ones).

SIMULA model

The SIMULA language proved its capacity to facilitate the design and the implementation of simulation models. It includes coroutines and processes of the discrete event simulation. Many existing classes extend the possibilities of the language. The Gpsss class provides the basic queueing network objects such as the facility (one server station), storage (resource, semaphore and stock), transaction (customer), region of statistics and automatic simulation report. It can be employed as in GPSS programming but with all the abilities of an object oriented simulation language.

Table 3: Partial report of the SIMULA model

```
* storages *
************
                 avg.    avg.time   contents
       entries contents  transit    now   max capa    util.
carriageL  13      1.00     36.78    0     2    2    49.81%
personnel  26      1.87     34.58    0     4    4    46.82%

* regions *
***********
            avg.     avg.     time      contents   non-zero
          entries contents  transit     now   max   transit
LoadL        13      0.21     7.80       0     2      7.80
MoveCL       13      0.41    15.15       0     2     15.15
UnloadL      13      0.21     7.79       0     2      7.79
MoveDL       13      0.16     6.05       0     2      6.05
```

The above table gives some results of the SIMULA programme. It contains the utilization rate of the personnel and of the linen carriages, the number of transactions carried out in the laundry and some average values and durations. Triangular and truncated Gaussian distributions have been tested, and they provide thinner confidence intervals. This model gives the highest priority to truck unloading by the logistic agents and it is more detailed than the following one.

WITNESS model

For the WITNESS action models, we tried to apply this industry tool to the hospital domain in defining a relation between knowledge model entities and WITNESS elements. The main flow elements are the missions, they may be represented by articles and they are characterized by two attributes: a mission kind and a transport mean. We use the following principle: the linen and carriages of linen are represented by articles entering in a stock then we represent the destination care units by machines, their processing times are defined by probability distributions according to table 1.

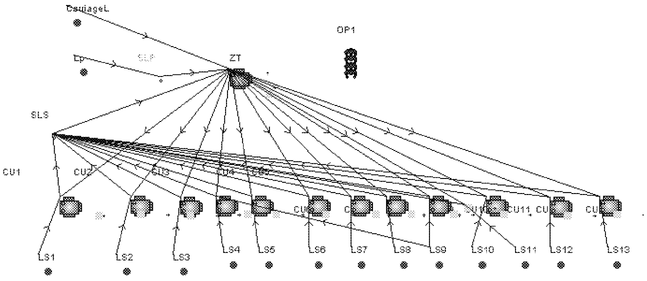


Figure 7: WITNESS model

When the truck arrives early in the morning, it deposits the linen bags represented by articles in the transit zone represented by a machine then the agents represented by resources (OP1) come to take the bags of clean linen in the transit zone then distributed to the different care units represented by machines with appropriate quantity (table 2), then the agents collect the dirty linen bags and return to the transit zone.

Table 4 contains the results of the WITNESS model. They concern the articles: a number of entries (Entries) an average response time (AvRT), and the resources: utilization rate (%Util) and the average utilization time (AvUT).

Table 4: Partial results of the WITNESS model

ARTICLE: CarriageL		RESOURCE: OPERATOR OP1	
Entries	13	%Util	47.76
AvRT	21.83	AvUT	21.33

Comparison of models

The results of the action models of the laundry concern specifically the personnel resource OP1. The table 5 presents: a minimal value (Min), a maximal value (Max), a mean value (Mean-value) and a confidence interval [Inf, Sup] at a 5% risk. Each simulation model provides 100 replications. Finally we can say that the WITNESS model provides a utilization rate ranging between the minimal value and the maximum of the SIMULA model. Of course, a lower risk gives larger intervals.

Table 5: Comparison of both models

OP1	SIMULA	WITNESS
Min	42.11	43.79
Max	50.04	52.8
Mean-value	45.64	46.41
[Inf, Sup]	45.24, 46.04	47.32, 48.49

CONCLUSION

We have introduced a methodology for hospital logistic modelling and simulation of the NHE laundry service. SIMULA is initially used for validation purpose of other simulation models but it is more efficient and accurate to model resources, stocks, delays and to more easily provide suitable statistics than WITNESS. The decision-making aid tool, realized with the Gpsss class, automatically determines confidence intervals for the utilization rate of resources and mainly for the agents of the NHE transit zone depending on various parameters. The WITNESS model provides a graphic interface with animation of the missions in each care unit. Finally this paper gives a first model version of the operation service laundry, with a perspective to study several management policies as well as using more realistic estimated data.

REFERENCES

André V. 2005. “Modélisation et simulation des flux logistique du nouvel hôpital d’Estaing”. Rapport technique, October 2005.

Benanteur, Y., R. Rollinger et J.-L Saillour. 2000. “L’organisation logistique et technique à l’hôpital”. Éditions ENSP, 185p.

Chabrol, M., P. Féniès, M. Gourgand et N. Tchernev. 2005. “Un environnement de modélisation pour le système d’information de la suply chain. Application au Nouvel Hôpital d’Estaing”, Grenoble Mai 2005.

Chen, P. 1976. “The entity relationship model - Toward a unified view of data”. ACM Transaction on data base system, March 1976.

Chauvet, J., N. Dessommes, N. Durand et A. Quinkal. 2005. “Un modèle de connaissance pour le nouvel hôpital d’Estaing”. DESS management, université d’Auvergne, Clermont-Ferrand, March 2005.

Combes, C. 1994. “Un environnement de modélisation pour les systèmes hospitaliers”. Thèse de doctorat, Université Blaise Pascal de Clermont-Ferrand, 27 October 1994.

Gourgand, M. et P. Kellert. 1991. “Conception d’un environnement de modélisation des systèmes de production”. 3ème congrès international de Génie Industriel, Tours.

Gourgand, M., F. Mebrek and A. Tanguy. 2005. “Hospital logistic modelling and simulation case study: brancardage”. ESM05, Porto, Portugal, 24-26 October 2005.

Green, P. and M. Roseman. 2000. “Modelling: An ontological evaluation”. In Information systems, ated process, vol. 25. 73-87.

Mebrek, F. et A. Tanguy. 2006. “Modélisation et simulation à événements discrets du pôle imagerie d’un hôpital moderne”. MOSIM’06 Maroc-Rabat 3-5 April 2006.

Sheer, A.W. 2002. ARIS - “Business Process Modelling”. Springer.

A Queueing Network Model of Patient Flow in an Accident and Emergency Department

S.W.M. Au-Yeung, P.G. Harrison and W.J. Knottenbelt
Department of Computing
Imperial College London, SW7 2AZ, UK
E-mail:{swa02,wjk,pgh}@doc.ic.ac.uk

KEYWORDS
Healthcare modelling, hospital logistics, A&E patient flow, queueing network model, discrete event simulation

ABSTRACT

In many complex processing systems with limited resources, fast response times are demanded, but are seldom delivered. This is an especially serious problem in healthcare systems providing critical patient care. In this paper, we develop a multiclass Markovian queueing network model of patient flow in the Accident and Emergency (A&E) department of a major London hospital. Using real patient timing data to help parameterise the model, we solve for moments and probability density functions of patient response time using discrete event simulation. We experiment with different patient handling priority schemes and compare the resulting response time moments and densities with real data.

INTRODUCTION

It is a goal universally acknowledged that a healthcare system should treat its patients – and especially those in need of critical care – in a timely manner. However, this is often not achieved in practice, particularly in state-run public healthcare systems that suffer from high patient demand and limited resources.

In the United Kingdom, there has been much public concern regarding patient waiting times in the National Health Service (NHS). For example, in a recent King's Fund report, improved waiting times for patients in Accident and Emergency departments and for cancer and cardiac patients are identified as two of the public's top four priorities for public healthcare in the UK [9].

In response, the UK government has introduced performance targets for the NHS, many of which are driven by response times – in 2004/2005 NHS performance ratings were based on eight key targets, six of which involved patient waiting and treatment times. Currently NHS Trusts are assessed against a broader set of core standards, but these still incorporate existing response time targets. For example, 98% of patients should spend 4 hours or less in an Accident and Emergency (A&E) department from arrival to admission, transfer or discharge. Although the vast majority of Acute trusts have managed to achieve a 95% threshold (assisted by innovations identified by the Emergency Services Collaborative such as "see and treat" schemes for minor injuries and near-patient testing [6]), 44% of Acute trusts are still failing to meet the 98% target [4]. This reflects the difficulty that many departments are experiencing in making further efficiency improvements [3]. This may be due, in part at least, to a lack of appropriate performance models and other systematic procedures for locating non-obvious capacity bottlenecks [6].

In this paper, we formulate a (simplified) hierarchical Markovian queueing network model of patient flow in the A&E department of a major London hospital. Using real patient timing data to help parameterise our model, we compute moments and densities of patient treatment time using a discrete event simulation. We investigate the impact of giving priority treatment to different classes of patients, and compare the resulting response-time densities and moments with real data. We believe this work represents an important initial step towards the creation of a formal modelling environment for patient flow in hospitals that will allow hospital managers to assess the response-time impact of different resource allocations, patient treatment schemes and workload scenarios, and thereby to implement optimised patient flow pathways.

The idea of modelling health service departments is, of course, by no means new. Several studies have been made of patient flow in hospitals in general [7, 8, 15] and Emergency departments in particular [1, 2, 11, 12, 13, 5, 14]. However, these studies have had limited success and subsequent impact for two main reasons. Firstly, there has been a lack of sophistication in the models used (mostly simple discrete event simulations and very high-level queueing models), and in the analysis techniques applied (mostly aimed at computing simple resource based measures such as utilisations and mean response times). Secondly, existing models frequently remain unvalidated using real waiting time data, since collecting this data was until recently a time-consuming, expensive, manual operation. We now have a prime opportunity to take advantage of the detailed patient waiting time data automatically collected by all A&E departments in England to monitor compliance with government targets (describing time of arrival, various treatment times and time of discharge for every patient). The remainder of this paper is organised as follows. The next section describes our multiclass Markovian queueing network model of patient flow. The numerical results section compares actual patient response times with our simulation results. The final section concludes and considers opportunities for future work.

QUEUEING NETWORK MODEL

Description

Figs. 1 and 2 show the simplified multiclass queueing network model of patient flow we have developed in conjunction with an A&E consultant at our case study hospital. The model takes the form of a hierarchical network of $M/M/m$ queues. Fig. 1 shows top-level patient routing with various aggregated servers; their corresponding lower-level expansions are presented in Fig. 2. Our queueing model has four customer classes: patients with minor illnesses or trauma (minors), patients with major illnesses or trauma (majors), patients requiring resuscitation (resusc) and patients that have yet to be classified (assessment). Customers can change class as they proceed through the system. In the top-level model there are two forms of patient arrivals: walk-in patients who come into A&E via their own transport and patients that arrive by ambulance.

Walk-in Patients

These patients enter via the A&E waiting room where they are registered at reception. The receptionists route each patient into one of three queues: patients with a clear case of minor trauma are placed in the minors queue; patients with a clear case of a serious illness or serious trauma are sent to the majors queue; all others (including all suspected cases of minor illness), are sent for nurse assessment.

Minors Queue Patients in the minors queue must first wait for a minors cubicle to become free; the patient then waits there for a minors practitioner (either a minors doctor or a nurse practitioner) to see them. The minors practitioner can decide to:

- Perform investigative tests and/or scans such as blood tests and x-rays, or
- Ask for a specialist opinion, or
- Treat (if necessary) and discharge the patient (to home, their General Practitioner (GP) or to the pharmacy to pick up medication), or
- Send the patient to be admitted to a (surgical) ward, or the Medical Assessment Unit (MAU) which assesses the need for medical admissions.

Majors Queue Patients in the majors queue wait for a bed in a majors bay to become free; once there, a nurse may perform tests (e.g. vitals, blood tests, x-ray) so that essential information is ready for a doctor. When the doctor has assessed the patient, (s)he may require a specialist opinion, request more tests, or send the patient out of A&E (possibly after treatment) via the routes mentioned above for the minors queue. Occasionally a patient may suffer a sudden and rapid deterioration, in which case the patient is transferred to a resuscitation bay and is attended to by the resuscitation team. Tests for both majors and the minors are processed in the same laboratory and radiology facilities.

Nurse Assessment Patients in the nurse assessment queue wait for an assessment room to become available; they then wait there for a nurse who assesses the severity of their illness or injury. The nurse can send the patient either to the minors queue, the majors queue or discharge them out of A&E to a specialist clinic, ward, GP etc.

Specialists Specialists may be called in by a minors practitioner or majors doctor. Minors patients are only referred to "other" specialists which encompass ENT, Gynaecology and Orthopaedics. Majors patients may be seen by medical, surgical and "other" specialists. After assessment, patients are discharged from A&E, either being sent to a clinic for a more thorough investigation, being admitted to a ward or being sent to the MAU.

Ambulance Arrivals

These patients are handed over to a nurse from the ambulance. The nurse assesses the patient, decides which queue to assign them to, and sends them either to reception to be registered or straight to a majors bay (as appropriate).

Blue Call Blue Call arrivals are very seriously ill or injured patients that require urgent medical attention. They almost always arrive by ambulance. Such patients are assigned a resuscitation bay and are attended to by a resuscitation team. Once stable the patient leaves A&E, being sent either to an operating theatre, to the Intensive Treatment Unit (ITU), or to a ward. Patients who cannot be resuscitated are sent to the mortuary.

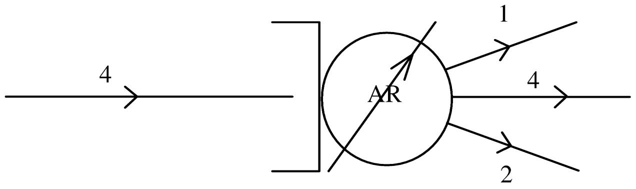
Passive resources

Note that, in many cases a patient needs to obtain a (passive) resource before they can progress along a treatment path. An example is the nurse assessment rooms (of which there are 5 in our A&E department). A patient must wait for one to become free before entering the room for assessment by a nurse. Once the assessment has been completed, the patient leaves the room,

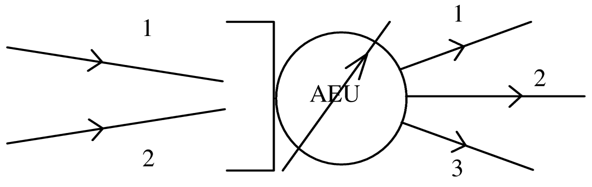
Patient Classes

1. Minors
2. Majors
3. Resusc
4. Assessment

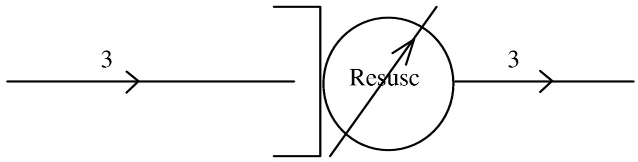
Aggregated Server AR (Assessment room)



Aggregated Server AEU (Whole medical unit)



Aggregated Server Resusc (Blue Call)



Top-Level Model

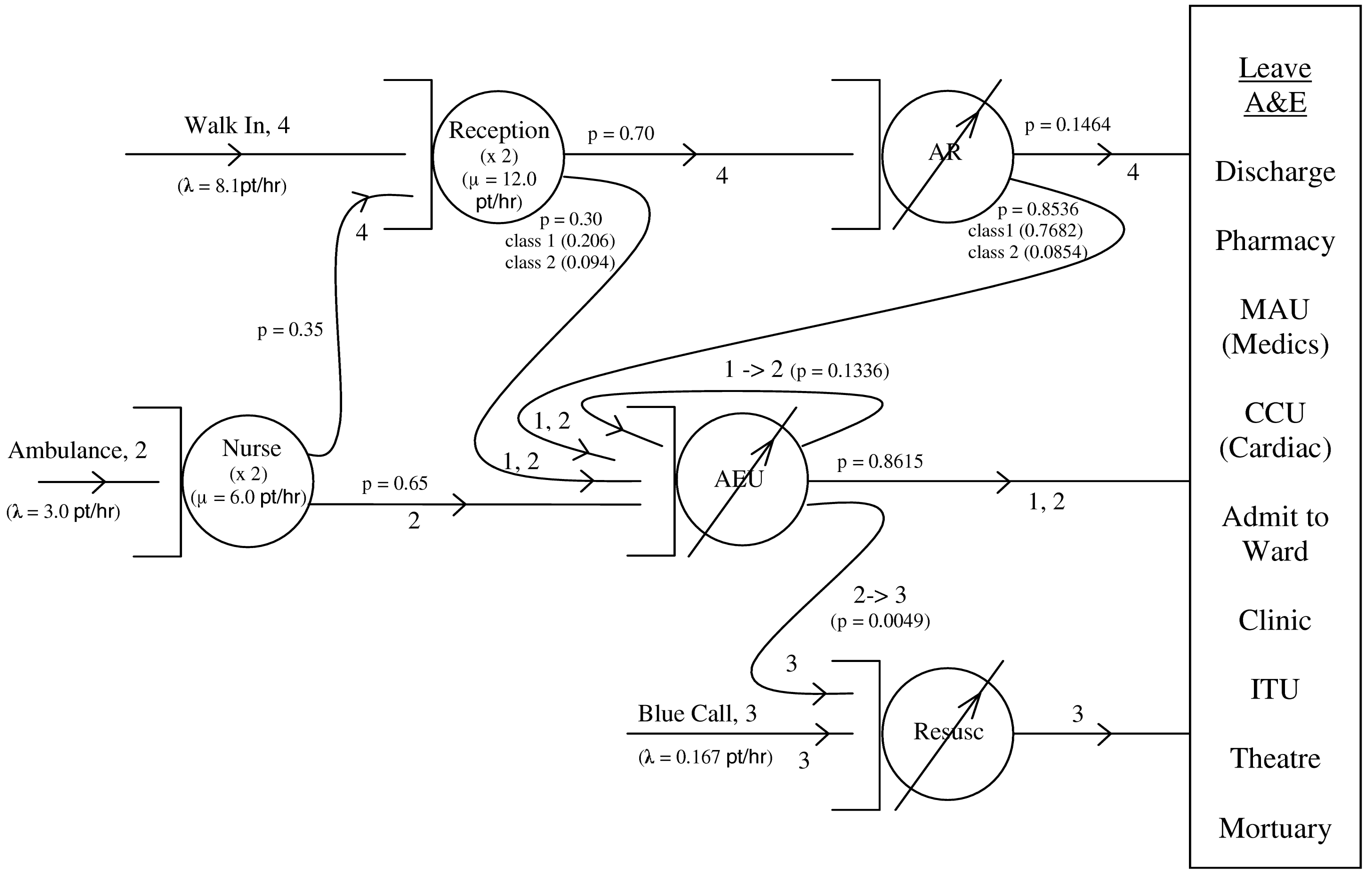


Figure 1: Top-level of queueing network model of patient flow

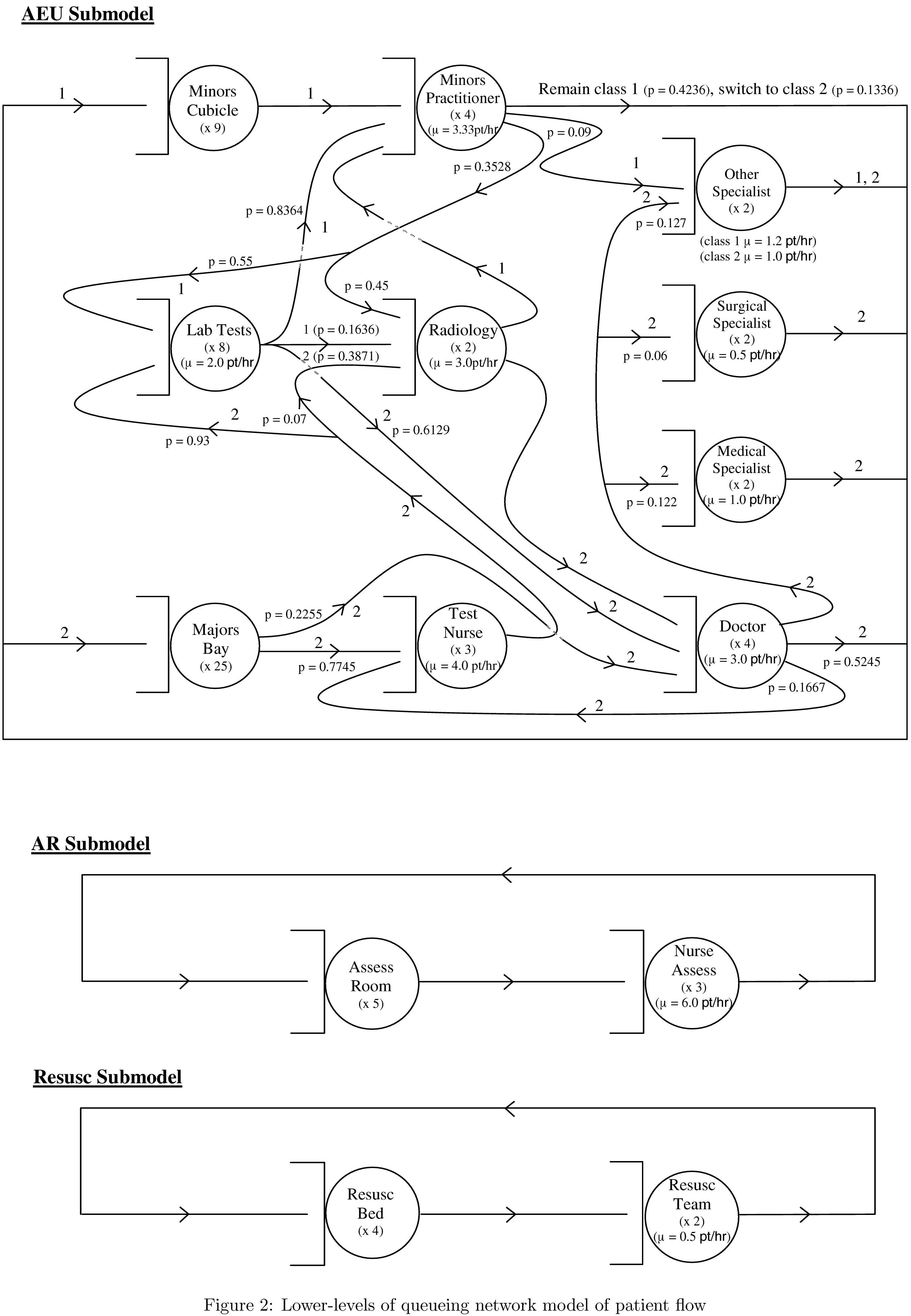

AEU Submodel
1
Minors Cubicle (x 9)
1
Minors Practitioner (x 4) (μ = 3.33pt/hr)
Remain class 1 (p = 0.4236), switch to class 2 (p = 0.1336)
p = 0.09
p = 0.3528
1
2
p = 0.127
Other Specialist (x 2)
(class 1 μ = 1.2 pt/hr)
(class 2 μ = 1.0 pt/hr)
1, 2
p = 0.8364
1
p = 0.55
1
1
p = 0.45
Lab Tests (x 8) (μ = 2.0 pt/hr)
1 (p = 0.1636)
2 (p = 0.3871)
Radiology (x 2) (μ = 3.0pt/hr
2
p = 0.06
Surgical Specialist (x 2) (μ = 0.5 pt/hr)
2
2
p = 0.93
p = 0.07
2
p = 0.6129
2
2
p = 0.122
Medical Specialist (x 2) (μ = 1.0 pt/hr)
2
2
2
2
2
Majors Bay (x 25)
p = 0.2255
2
2
p = 0.7745
Test Nurse (x 3) (μ = 4.0 pt/hr)
2
2
Doctor (x 4) (μ = 3.0 pt/hr)
2
p = 0.5245
p = 0.1667
2
AR Submodel
Assess Room (x 5)
Nurse Assess (x 3) (μ = 6.0 pt/hr)
Resusc Submodel
Resusc Bed (x 4)
Resusc Team (x 2) (μ = 0.5 pt/hr)

Figure 2: Lower-levels of queueing network model of patient flow

freeing it up for the next patient. Other passive resources include minors cubicles (of which there are 9), majors bays (of which there are 25) and resuscitation beds (of which there are 4).

Complexities not modelled

In a real-life A&E unit there are many additional complexities that we have not incorporated into our model. For example, patients who should be discharged from A&E to another hospital ward are sometimes held there even though their treatment in A&E is complete because there is no bed available for them in the destination ward. Similarly, patients may be held in A&E after completion of treatment awaiting an ambulance to take them home. We have not modelled these blocking phenomena caused by factors outside the A&E unit.

Patients who cannot walk must be transported around the A&E unit and taken to other areas of the hospital by porters; these are not represented in our model.

We have also had to simplify the nature of the tasks undertaken by various staff. For example, we have assigned nurses to perform specific tasks e.g. some nurses only assess patients. In a real A&E unit all the nurses are trained to perform assessments and treatments and so provide a more flexible staffing pool. As another example, there are many more types of specialist available in a real hospital than we have modelled. Also staffing levels and patient arrival rates vary throughout the day; we have used average values in order to simplify our model.

Finally, we have incorporated treatment time into the time it takes for a patient to be seen by either the doctor or minors practitioner. Depending on the nature of the patient's illness or injury, this may or may not be the case in an actual A&E unit.

Parameterising the model

We have obtained ethical approval to access detailed patient timing data collected by our case study A&E department in a North London hospital. Where possible, we have used this data to parameterise our model. In particular, data for the year April 2004 to April 2005 was used to work out mean arrival rates: in that year there were 70 909 walk in arrivals and 26 285 ambulance arrivals; from experience there are 4 blue call arrivals a day – giving us mean arrivals rates of 8.1 walk in arrivals per hour, 3.0 ambulance (but not blue call) arrivals per hour and 0.167 blue call arrivals per hour. Where possible, we used the data to derive patient routing probabilities and mean service times; where this was not practical, we have used estimates provided by an A&E consultant, who has also checked the patient flows. Staff and resource numbers were provided by the hospital. Since there are different staffing levels throughout the day, we have taken average values (see Figs. 1 and 2 for staff numbers and service rates).

NUMERICAL RESULTS

We now compare numerical results from our discrete event simulation (written in Java) and real data.

Mean and variance of patient response time

Table 1 compares the first two (central) moments of patient response time for various types of patient arrival (Walk-in, Ambulance and Blue call arrivals) as calculated using our discrete-event simulation. The simulation results presented are the average of ten runs. Each run includes a transient period during which 2 000 000 patients move through the system (and during which passage time statistics are not collected), followed by a measurement period which lasts long enough to observe 10 000 passages of Blue Call arrivals through the system; in this period around 485 000 passages of Walk-in arrivals and 180 000 passages of Ambulance arrivals are also observed.

Three different patient priority schemes are analysed:

- *No Priority* in which First In First Out (FIFO) queues are implemented,
- *Majors Priority* in which majors patients are given priority at the shared resources (lab tests, radiology and "other" specialist), and
- *Minors Priority* in which minors patients are given priority at the shared resources.

From Table 1 it can be seen how giving priority to the majors class seriously degrades the waiting time of the walk-in patients (in terms of mean and variance), which are predominantly minors. By contrast it might appear that seriously injured or ill patients arriving by ambulance actually benefit from giving minors priority. In fact both ambulance and walk in arrivals under minors priority are seemingly treated quicker than even a no priority system. However, this interpretation may be misleading: a significant proportion of ambulance arrivals end up as minors (about 35%) and their benefit outweighs the penalty suffered by the majors that arrive by any means. Conversely, the walk-in major patients are highly penalised because relatively few walk-in minors patients switch to majors (about 16%). A separate comparison of ambulance arrivals that are treated as majors throughout their stay against walk-ins that are treated as minors throughout, i.e. neglecting any patients that change class, will reveal the true effects of changing between majors and minors priority. However, it must be remembered that the most important statistics to the individual patient concern their own time spent in hospital, regardless of the class to which they may be assigned.

Table 2 shows the first two moments of patient response time for various types of patient arrival (Walk-in, Ambulance and Blue call arrivals) as actually observed in

	Walk-In arrivals		**Ambulance arrivals**		**Blue Call arrivals**	
	E[T]	Var[T]	E[T]	Var[T]	E[T]	Var[T]
No Priority	2.86	8.57	2.77	5.28	2.08	4.19
Majors Priority	5.15	37.22	3.48	17.19	2.06	4.12
Minors Priority	2.05	4.05	2.63	4.82	2.07	4.15

Table 1: Mean and variance of response times (in hours) for walk in, ambulance and blue call patients under major, minor and no priority schemes.

	Walk-In arrivals		**Ambulance arrivals**		**Blue Call arrivals**	
	E[T]	Var[T]	E[T]	Var[T]	E[T]	Var[T]
2002/2003	3.22	13.03	5.69	23.40	4.18	26.95
2003/2004	2.46	4.98	4.22	9.73	2.43	4.81
2004/2005	2.04	2.54	3.14	4.49	2.09	3.37

Table 2: Observed mean and variance of response times (in hours) for different classes of arriving patient.

the A&E we have modelled. Figures are reported over three annual reporting periods (2002/2003, 2003/2004 and 2004/2005), where each reporting period begins on 1 April and ends on 31 March the following year (coinciding with the hospital's financial year). One can readily observe the effect of the introduction and subsequent tightening of patient response time targets. The practical effect of this has been to move from a system in which majors are given priority treatment to a system in which minors are (to a large degree - since the majority of patients in A&E departments are minors patients) given priority treatment. Indeed, the trends observed (with associated reductions in the mean and variance of patient waiting time) are consistent with those we observe when moving from a majors priority to a minors priority schemes (cf. Table 1).

When comparing mean patient response times from our minors priority simulation model with the observed 2004/2005 figures, we observe differences of 0.5%, 16.2% and 1% for Walk-in, Ambulance and Blue call patients respectively. The relatively large disagreement between ambulance arrival actual treatment time and our simulation may be due to the lack of blocking phenomena in our model, which will mostly delay ambulance arrivals. However, the close agreement for Walk-in and Blue call arrivals is promising, considering the many simplifying assumptions we have made.

Densities of patient response time

Figs. 3, 4 and 5 show the simulated vs. actual patient response time densities for Walk-in, Ambulance and Blue call arrivals respectively; note that the curves corresponding to the no priority system lies in between the curves for the majors and minors priority systems, also note the peaks in the 2004/2005 actual patient response time densities corresponding to the four hour target.

CONCLUSION AND FUTURE WORK

In this paper, we have used a simulation model to provide some insights into the effects of prioritising different classes of patients in a real A&E unit. We have found that the (seemingly socially unacceptable) prioritisation of treatment for minors (i.e. patients with minor illness or trauma) over majors (i.e. patients with severe illness or trauma) can lead to the counter-intuitive outcome that mean response times for ambulance arrivals are not adversely affected (in fact they are slightly improved), while mean response times (and corresponding variances) for walk-in arrivals are dramatically lower. This is a particularly interesting result in light of UK government waiting time targets, which encourage the prioritisation of minors.

In the future, we intend to incorporate more realistic assumptions into our models. For example, the arrivals process at a real hospital is non-stationary and is more bursty than a Poisson arrivals stream. Also, our model does not yet represent the "rising panic" phenomenon that occurs in real A&E units whereby patients are subject to higher and higher priority treatment as they approach the four hour waiting time target. Some progress towards modelling queues where the priority of a customer increases with the time spent in-queue was made by one of the present authors in [10]. There a queue was represented by an ordered set of current customer *sojourn times*. This has led to a uniform way of deriving response time distributions under various queueing disciplines and a rather complex, untested approximate route to deadline queues. We will tailor this approximation to our problems.

ACKNOWLEDGEMENTS

We are grateful for the help and advice of many members of staff at our case study hospital and associated institutions, including John Knottenbelt, Rick Juniper,

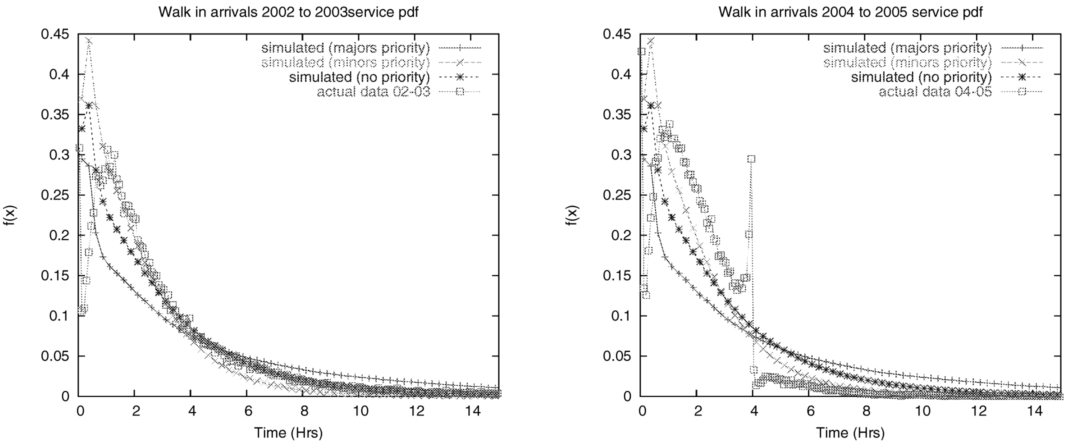


Figure 3: Actual and simulated response time density for walk-in arrivals using 2002/2003 data (left) and 2004/2005 data (right)

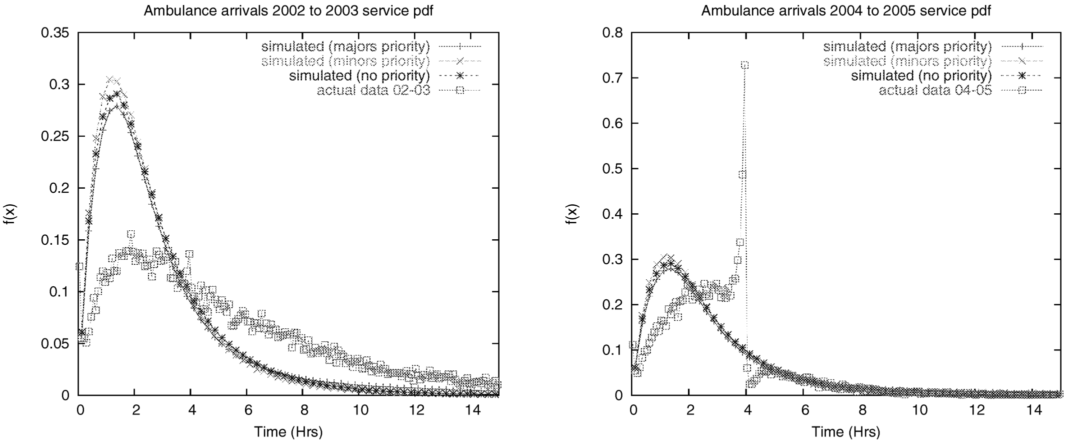


Figure 4: Actual and simulated response time density for ambulance arrivals using 2002/2003 data (left) and 2004/2005 data (right)

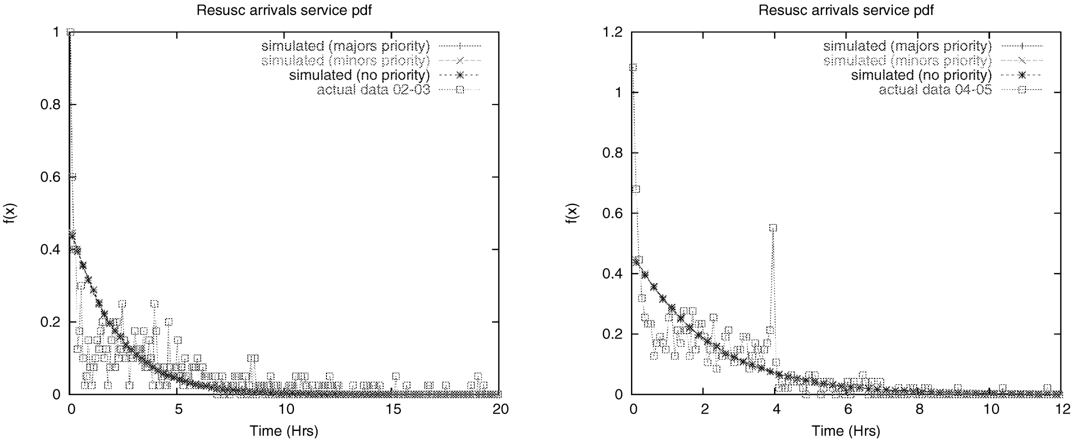


Figure 5: Actual and simulated response time density for blue call arrivals using 2002/2003 data (left) and 2004/2005 data (right)

David King, Raj Singh, Sunil Johal, Sharon Ahearn and Ken Walton. We would also like to thank Tony Field for the use of his JINQS Java queueing network simulation library. Ethical approval for access to psuedonymised patient records was granted by the Harrow Local Research Ethics Committee (Ref. 04/Q0405/72).

REFERENCES

[1] J.T. Blake and M.W. Carter. An analysis of Emergency Room wait time issues via computer simulation. *Information Systems and Operational Research (INFOR)*, 34(4):263–273, November 1996.

[2] T.J. Coats and S. Michalis. Mathematical modelling of patient flow through an Accident and Emergency department. *Emergency Medicine Journal*, 18:190–192, 2001.

[3] Healthcare Commission. Acute hospital portfolio review. accident and emergency. Technical report, August 2005.

[4] Healthcare Commission. NHS performance ratings 2004/05. Technical report, 2005.

[5] L.G. Connelly and A.E. Bair. Discrete event simulation of emergency department activity: A platform for system-level operations research. *Academic Emergency Medicine*, 11(11):1177–1185, 2004.

[6] M. Cooke, J. Fisher, and J. Dale et al. Reducing attendances and waits in emergency departments a systematic review of present innovations. Technical report, Report to the National Co-ordinating Centre for NHS Service Delivery and Organisation R & D (NCCSDO), 2005.

[7] Murray J. Côté and William E. Stein. An Erlang-based stochastic model for patient flow. *Omega: The International Journal of Management Science*, 28:347–359, 2000.

[8] R. Davies and H.T.O. Davies. Modelling patient flows and resource provision in health systems. *Omega: The International Journal of Management Science*, 22:123–131, 1994.

[9] The King's Fund. Has the government met the public's priorities for the NHS?: A King's Fund briefing for the BBC 'Your NHS' day 2004. Technical report, 2004.

[10] P.G. Harrison. An M/M/1 queue with aging priority. In *Proc. International Conference on Stochastic Modelling and the IV International Workshop on Retrial Queues*, Cochin, India, December 2002.

[11] D. Lane, C. Monefeldt, and J. Rosenhead. Emergency – but no accident – a systems dynamics study of an Accident and Emergency department. *OR Insight*, 11:2–10, 1998.

[12] L. Mayhew and E. Carney-Jones. Evaluating a new approach for improving care in an Accident and Emergency department: The NU-care project. Technical report, Cass Business School, City University, 2003.

[13] Ò. Miró, M. Sánchez, G. Espinosa, B. Coll-Vinent, E. Bragulat, and J. Millá. Analysis of patient flow in the emergency department and the effect of an extensive reorganisation. *Emergency Medical Journal*, 20:143–148, 2003.

[14] A. C. Virtue. Simulating accident and emergency services with a generic process model. Nosokinetic News, December 2005.

[15] E.N. Weiss, M.A. Cohen, and J.C. Hershey. An iterative estimation and validation procedure for specification of semi-Markov models with application to hospital patient flow. *Operations Research*, 30(6):1082–1104, 1982.

AUTHOR BIOGRAPHY

SUSANNA AU-YEUNG is a Ph.D student in the Department of Computing at Imperial College London where she is working on characterising and modelling patient flow into and within A&E departments. From these models, performance measures and densities are extracted and analysed.

PETER HARRISON is a Professor of Computer Science in the Department of Computing at Imperial College London. He has researched into quantitative methods for many years, in particular queueing theory, stochastic modelling and their applications in performance engineering. His work ranges from novel theoretical concepts (e.g. negative customers and Stochastic Process Algebras) to specific applications in the prediction of the performance of parallel and distributed systems. His research has been supported by several grants from the EPSRC and EU.

WILLIAM KNOTTENBELT is a Senior Lecturer in the Department of Computing at Imperial College London. His research work has focused on the performance modelling of concurrent systems using high-level modelling formalisms such as stochastic Petri nets, stochastic process algebras and queueing networks. Most recently, he has developed algorithms for computing response time densities and quantiles. His research has been supported by two research grants from the EPSRC.

A CAPACITY PLANNING SIMULATION MODEL AND ITS APPLICATION TO A NUCLEAR MEDICINE SERVICE.

Rob Cameron
Director
Lattice Networks Limited
Kings Langley
WD4 8DG
rob@lattice-net.co.uk

Robert E Dugdale
Foundation Trust Project Manager
Bradford District Care Trust
Bradford
BD9 6DP
bob.dugdale@bdct.nhs.uk

Michael J Page
Nuclear Medicine Section Head
Department of Medical Physics
Bradford Teaching Hospitals, Yorkshire
BD9 6RJ
mike.page@bradfordhospitals.nhs.uk

KEYWORDS
Resource management, capacity planning, healthcare, nuclear medicine, modelling, simulation.

ABSTRACT

A capacity planning simulation program was developed to assist in the management of the diagnostic nuclear medicine service in a large teaching hospital in the north of England. The application is configurable and could be applied to the modelling and simulation of other services. The aim was to provide a tool for managers to make informed decisions about resource utilisation with the objective of reducing waiting times for patients requiring the service. It has also been used to estimate the changes in work patterns necessary if the service is expanded. The paper describes the architecture of the application, its modes of operation and presents a preliminary study. An outline of future development is given.

INTRODUCTION

The nuclear medicine facility at Bradford Teaching Hospitals NHS Trust (BTHNHST) has two Gamma cameras located in the Department of Medical Physics, one with Positron Emission Tomography (PET) capability. In the 12 months to April 2003, 3835 patients were scanned. Forty-five different scanning procedures were used, some of which required follow-up scans either the next day or at longer intervals. Some scans require multiple follow-ups. The department serves a large area of West and North Yorkshire with a population of over 600 000. Four technical staff routinely work in the nuclear medicine section, augmented from time to time by *locum* staff.

Scheduling patients from the waiting list into weekly timetables is a demanding task because of the complex timing involved in the procedures and difficulty of assessing priority order in the waiting list. This task is currently carried out manually by one of the senior technical staff.

A recently completed development of the BTHNHST included three new surgical wards and a possible future expansion of the geographical referral area and this emphasised the need for powerful management planning tools (Austin and Boxerman, 2003). The development of a capacity planning model was commissioned, with the brief of being able to simulate the current service with reasonable fidelity, to assist with managing resource utilisation to reduce waiting times and to provide estimates of the service changes needed to accommodate forthcoming increases in demand.

Some beneficial side-effects of the model-building procedure were anticipated: further systematic development of the process (e.g. rationalising codes which refer to procedures) and a software tool for time-tabling patients.

THE PROCESS TO BE MODELLED

The first step was to produce narrative and process flow models of the operation of the nuclear medicine section. This was considerably facilitated by a concurrent initiative to achieve ISO 9000:2000 certification.

Both out-patients and in-patients are referred to the service by a consultant, a primary care trust or another hospital. Patient studies are scheduled on a weekly basis about three weeks in advance. In this period, patients are notified and invited to attend at an appointed time. At their appointment, a patient will typically be given an injection of pharmaceutical material and asked to wait for a specified period of time (the ‘latency’). There will then be time spent while the gamma camera completes a scan. After the scan, the images are viewed by a radiologist and a report generated within 48 hours.

There are many constraints on the process, of which a sample are:

- ξ Only appropriately qualified staff can administer injections;
- ξ Patients must be accompanied by a member of staff while they are being scanned in the camera (for paediatric scans, two members of staff must be present during the injection phase);
- ξ Some procedures must be carried out using a specified camera.
- ξ Some scans often need to be repeated;
- ξ Some procedures are required at short notice and so are not pre-booked. A certain number of sessions are allocated each week for a ‘drop-in’ service.
- ξ Some scans must be completed within a short period of the request being made.

- ξ Some tests are in two parts, e.g. an exercise stage and a resting stage. These are usually scheduled for the same week, but on different days.

All patients must be seen within 6 months of referral.

THE CAPACITY PLANNING MODEL

Choice of simulation environment

The process is essentially a multiple queuing process (one queue for each procedure) with finite, cost-limited resources for processing the queues. Following current scheduling practice, it was decided to work on a discrete-time interval of one week. Several commercial simulation tools are available for discrete-time simulation, and some examples of their use in clinical management have been published (see for example: Lowery (1996), Price and Harrell (1999), Standridge 1999). While these systems undoubtedly work well in some settings, the nuclear medicine process involves such complex constraints on the scheduling process that it was decided to write a dedicated simulation tool. This would also give the flexibility to include at a later date, such additional tools as fuzzy reasoning and, possibly, integration with the patient management system in use in the department.

Program architecture

The program architecture is shown in Fig. 1.

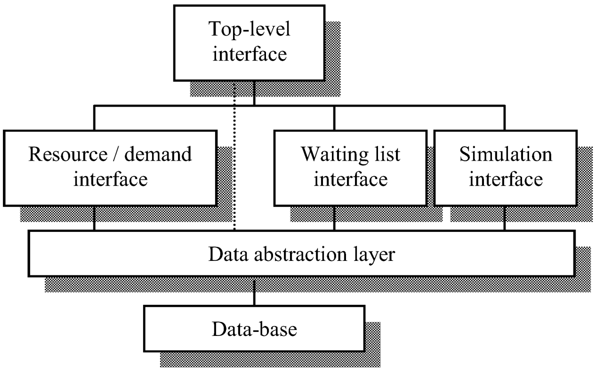


Figure 1: Architecture of the application

The program is written in Borland Delphi and uses the Firebird database (Borrie, 2004). In its current form it is intended to be a desktop application, but could easily be converted to networked client/server operation. Although the application was designed for the nuclear medicine process, it is highly configurable and could be applied to other resource management problems which have a similar underlying structure.

Top-level interface

The top-level interface manages user authorisation and logon, basic initialisation and access to the other modules via menu options.

Resource / demand interface

It is assumed that three types of resources are required: personnel ('*staff*'), room space and equipment. Each resource has two configuration tables. The first is used to define the range of resource types which are utilised, the second defines the actual instances of those types. For example there are two *staff types*, namely those who are, and those who are not, qualified to administer injections. Once these two types are defined, the actual staff list can be prepared, showing how many of each type are available. Similarly, there are four *Room types*, an injection room, an exercise room, a camera room and a camera room with PET. In the preliminary studies reported here, no equipment (e.g. accessories for the cameras) was defined because it is not capacity limiting, but in future studies it will be included. For each instance of a resource, the actual hours which are dedicated to the process being studied can be set. So, for example, the senior staff responsible for managing the service do not devote as many hours to scanning patients as do other colleagues.

The demand is set by detailing each procedure to be carried out. The many parameters include: times taken for injection, scanning, minimum and maximum latency and whether there is a follow-up procedure; the frequency of occurrence, a weighting factor which is used to assign the urgency of the procedure and the maximum permissible time a patient can be allowed to wait on the list.

Waiting list interface

This module allows the user to initialise, view and edit two lists – live and simulated. The live list allows reference to the actual patients waiting for appointments. This could be integrated into the patient management process.

Simulation interface

The simulation interface controls the configuration and progress of the simulations. Default simulation configurations are retrieved – or built – from information saved in the database in the resources/demand module. The number and availability of resources, and the level of demand can all be modified before a simulation run and each such 'scenario' stored in the database for future use.

The outputs of the simulation include graphs of the available resources (discounted to allow for staff breaks, camera downtime, etc) and of the percentage utilisation of the resources. The numbers of patients seen, the number on the waiting list at the end of each week and the approximate waiting time are also shown. Tabular results include the numbers of each of the procedures completed week by week, and the totals.

All the usual graphical features are available: rescaling, zooming, scrolling, etc.

Two modes of simulation are provided: 'simplified dynamics' and 'detailed scheduling'. In the former, certain assumptions are made which allow a faster, but slightly less accurate, simulation of the process, with no information about exactly when during a week a particular procedure has been scheduled. The key assumptions relate to those resources which are *limiting* (e.g. it is assumed that there is never a shortage of staff qualified to administer injections) and to the availability of rooms for follow-up procedures.

For example it is assumed that a procedure that needs a follow-up can always be booked; in real-life this is not always the case: sometimes such procedures must be postponed for a week. Only overall resource drain is summed; no account is taken of whether the patterns of each scan can in fact be 'tessellated'. However, the opinion of the technical staff who manage the service is that these are indeed reasonable assumptions and exceptions will in any case be infrequent. The advantage of this approach is that the time taken for a simulation run is short: a few seconds for a simulation of a few years. Thus the turn-round time testing proposed management strategies is short and several can be thoroughly explored in an hour or two.

In 'detailed scheduling' most of the timetabling constraints will be incorporated and a detailed schedule produced for each week. This approach has a considerably higher computational burden and simulation runs will take correspondingly longer. This development is on-going and in due course, this is expected to be a useful practical tool for drawing patients from the waiting list and planning the sequences of scans to be undertaken.

WAITING LIST MANAGEMENT

Building a virtual waiting list, and the process of drawing subjects from it are critical parts of the simulation. The likelihood of each procedure occurring per week was calculated from historical records and is used – in conjunction with a random number generator – to build the waiting list. If for example, a procedure was found to be required 4.3 times per week (on average) then, when building the list for a week: 4 patients are added; a random number, R ($0 \leq R \leq 1$), is generated; if $R < 0.3$, a fifth patient is added. When this is repeated for all procedures over the length of a simulation run (typically 2 – 5 years) the results average out to match closely the historically observed figures.

Patients are drawn from the top of the waiting list after it has been *prioritised*. A priority value is assigned to each patient on the list, and the list is sorted in decreasing order of priority. The priority value, P, is calculated as follows:

P = (number of days the patient has already been on the list) × (urgency weighting factor)

If a patient has been on the list for more than a specified fraction (default: 80%) of the maximum permitted waiting time, then 10000 is added to P; (this effectively pushes the patient to the top of the list, although if there is more than one such patient, there will still be an ordering proportional to their actual wait time and the urgency of the procedure. To resolve any priority values which may be equal at this stage, a small (< 1) random number is added to P. This procedure mimics – more or less – that used in practice.

VALIDATION

The simulation model was validated in two ways. First, the member of staff most familiar with the routine management of the nuclear medicine section executed it under many different conditions and compared it against his own intuition and experience. This approach could be expanded and made more rigorous in future work (see "Future developments" below).

Secondly, the numerical results generated by the simulation have been compared with historical records, using similar initial conditions. The results were satisfactory, but more work needs to be done, both to make the comparisons more rigorous and to use more challenging test data. Ideally, the program would be configured to match a similar service elsewhere and the outcomes assessed.

However, the model does not aim to mimic precisely the historical records, only to demonstrate average capacity and resource utilisation for specified resources and demand levels. The main purpose (as noted above) was to provide insight into the capacity limiting resources and explore viable strategies for expansion.

Several stochastic factors make precise validation difficult. Two of the most important are staff absences (annual leave, sickness, etc) and patients who do not attend (DNA). It is not our intention at this stage to incorporate detailed data about staff absence, only to estimate a discount percentage factor to reflect it. For example, over the course of a year annual leave and bank holidays account for approximately 10% of the working life of a member of staff. The model makes assumptions (clearly not realistic) that this absence is uniformly distributed throughout the year and (more realistically) that there aren't any periods of time when staff absences make staff availability the limiting resource. The assumptions are supported by the fact that current practice is to cover staff absence using staff overtime, or *locums* if the absence is lengthy. The number of missed appointments is around 10% but the impact on the service varies more than this implies. Some of the missed appointments need to be rebooked, so the patient is put back on the waiting list; sometimes the request is withdrawn. If a complicated procedure is missed (say with several follow-up appointments) the schedule is disrupted more than if the missed procedure is a simple one.

Other resource outages imply more significant disruption of the service. For example cameras are occasionally unavailable due to regular maintenance or breakdown. Regular servicing is predictable and (in real life) procedures are scheduled taking these events into account. On the other hand, unexpected breakdowns imply cancellation and rescheduling of scans at short notice with consequences for service provision that are difficult to predict.

Some exceptional events such as camera breakdown can provide an opportunity to challenge the model providing that detailed concurrent data is available.

EXAMPLE

As an example of the use of the program, a simulation was configured which approximately matches the service at BTHNHST. That is to say, 2 cameras, 4 members of staff and an average patient throughput of some 80 patients per week. The simulation models a hypothetical period of three years.

Figures 2 - 9 show respectively the available staff resource (person-hours per week) and the percentage utilisation of the staff; the camera resource and its utilisation; the variation in demand; the number of patients seen each week, the number of patients on the waiting list at the end of each week and the approximate waiting time at the end of each week.

Initially there were 400 people on the waiting list. In weeks 1 – 25 approximately 86 patients per week were added to the list and the same number were seen in the section, so the number of patients on the list, and the approximate waiting time were in equilibrium.

In weeks 25-27, a camera failure halved the available camera resource (Fig. 4), causing the waiting list to grow and dramatically increasing the estimated wait time (Fig. 9). In week 46 there was a 10% increase in demand (Fig. 6), which caused further increase in the waiting list. At week 60, the number of hours in the working week was increased (Fig. 4), to maximise the potential staff resource utilisation; this slowly reduced the queue. At week 100 additional staff resource was made available, but since the camera utilisation was at 100% throughout, this made no significant impact on either the waiting list or waiting time. At week 120, the working week was again increased, increasing capacity and markedly reducing the waiting list.

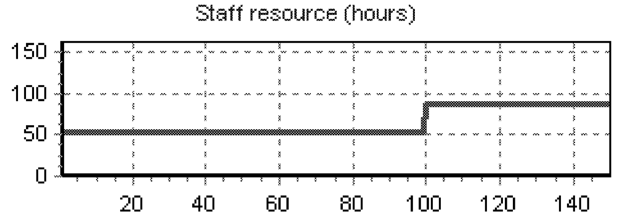


Figure 2: Available staff resource

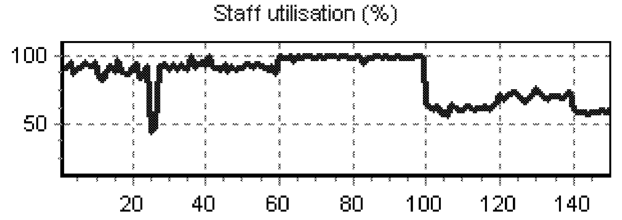


Figure 3: Staff utilisation (%)

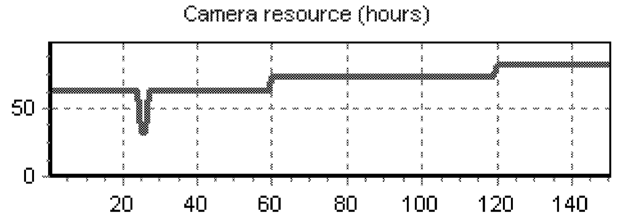


Figure 4: Available camera resource

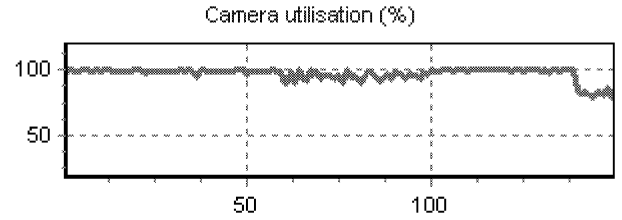


Figure 5: Camera utilisation (%)

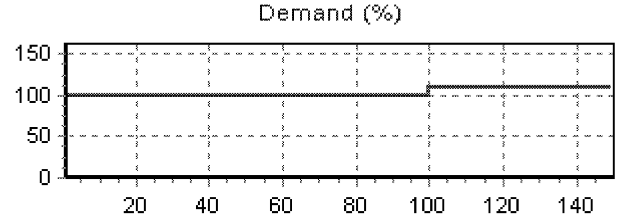


Figure 6: Demand factor

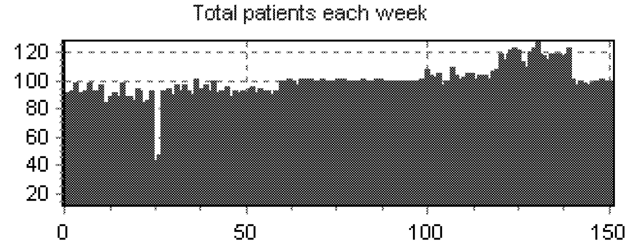


Figure 7: Patients seen per week

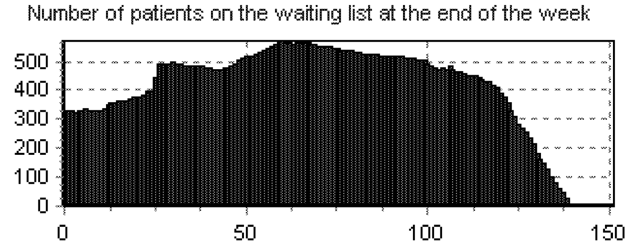


Figure 8: Length of list per week

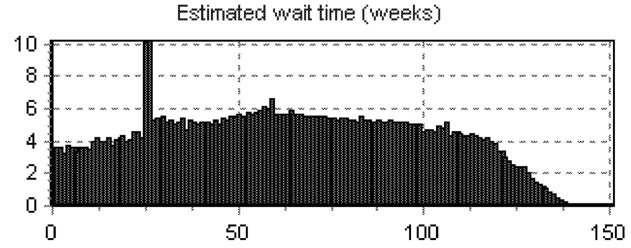


Figure 9: Approximate waiting time

It is noticeable that throughout this simulation 100% of the camera capacity was utilised (apart from the period when the waiting time fell to zero), suggesting that camera availability is the limiting resource in this example. In general, the limiting resource will be affected by the frequency distribution of the procedures requested by referrers. For example, increases in child referrals would put extra load on the staff resource while increases of referrals for procedures that require pre-scan exercise would increase demand for space (room) resources.

FUTURE DEVELOPMENTS

For the simulation, patients are currently drawn strictly in priority order from the waiting list. In practice, the selection is more complicated and takes into account a number of other factors. In fact, probably the most challenging aspect of the scheduling problem is the very long list of constraints which are applied to the timing of procedures and the number of factors which are taken into account in assigning a patient to a position in the waiting list. In the current version of the application, many of these factors and constraints are ignored. No attempt is made, for example, to schedule patients so as to minimise the pharmaceutical cost; this is taken into account by the member of staff who currently does this job manually. In future versions, it is hoped to incorporate more of these factors, perhaps using fuzzy reasoning.

It became apparent during the process analysis that several unexpected informal feedback mechanisms are at work. For example, if the waiting list for a particular procedure is getting 'too long' then the perceived urgency of this procedure will be increased by staff who timetable the work and more patients waiting for this procedure will be scanned to maintain a 'suitable balance' of the list. This appears to make sense in terms of resource utilisation, because by block-booking a series of patients for the same procedure, some savings can be made in the preparation of pharmaceuticals and the absence of the need to reconfigure the gamma cameras. Also, if the waiting list lengthens, extra working hours are funded to reduce the list.

A second unexpected feedback phenomenon affects the referral rate for each procedure. In the existing model these rates are assumed to be independent variables. In fact some referrers are aware of waiting times and alter their referring strategy. For example if the waiting time for a particular procedure increases, a referrer might make the judgement to directly refer a patient to a more complex or more expensive procedure without waiting for the preliminary result. This judgement is difficult to predict, since it will depend on the assessment of the patient's immediate condition, the costs of various procedures, the risks associated with any delays and the potential benefits of early diagnosis. Thus the demand is loosely coupled to the waiting time for specific procedures.

In the future these kinds of phenomena could be incorporated in the model, again probably using a fuzzy-control approach.

Disruption to the service due to camera downtime can have a significant effect on the number of patients on the waiting list and the corresponding waiting times. In the current model this is accounted for only in general terms by applying an attrition factor to the availability. Thus a certain percentage of camera time is assumed to be unavailable due to maintenance. This is currently (and falsely) assumed to be evenly distributed throughout the working year. Historical data on breakdowns is available in the section and more realistic patterns of breakdown could be incorporated in the model.

Finally, we would like to carry out more extensive model validation exercises, including using a modified 'Turing test' approach, where an expert would be presented with historical records and simulation results and invited to separate them into piles which s/he consider to be 'real' and 'simulated' respectively (Schruben, 1980). The model has higher credibility if its results are identified no more often than chance would dictate.

REFERENCES

Austin, C.J. and Boxerman, S.B. 2003. "Information Systems for Healthcare Management", (6th ed) Health Administration Press.

Borrie, H. 2004. "The Firebird Book", Apress. ISBN 1-59059-279-4. See also: http://firebird.sourceforge.net.

Lowery, J.C. 1996. Introduction to simulation in health care, *Proc. 1996 Winter Simulation Conference*, (Coronado, CA). ACM Press.

Price, R.N. and Harrell, C.R. 1999. "Healthcare Simulation Modeling And Optimization Using Medmodel," *Proc. 1999 Winter Simulation Conference* (Phoenix, AZ. Dec. 5-8) ACM Press.

Schruben, L. W. 1980. "Establishing the credibility of simulations", *Simulation*, **34**, pp. 101 – 105.

Standridge C. R 1999. "A Tutorial On Simulation In Health Care: Applications And Issues" *Proc. 1999 Winter Simulation Conference* (Phoenix, AZ, Dec. 5-8). ACM Press.

AUTHOR BIOGRAPHIES

ROB CAMERON is Director of Lattice Networks Limited, a small private company specialising in consultancy and software for SMEs and healthcare. His fields of interest are healthcare applications of modelling, simulation and database systems. He was formerly Senior Lecturer in Control Engineering at the University of Bradford and Research Co-ordinator in Medical Physics at Bradford Teaching Hospitals. In 1995/96 Rob held an EU "Marie Curie Fellowship" at the Dept. of System Engineering and Automatic Control in Valladolid, Spain. He is co-author of "Introduction to Mathematical Control Theory" (OUP) and was joint winner of the 1999 IEE 'Crompton Premium' for research on the control of power systems. He is a Chartered Engineer, a Clinical Scientist and a Senior Member of the IEEE.

BOB DUGDALE is Foundation Trust Project Manager at Bradford District Care Trust. He was formerly Director of Risk Management, Director of Clinical & Scientific Support Services and Head of Medical Physics at Bradford Teaching Hospitals NHS Trust.

MIKE PAGE is Head of the Nuclear Medicine section of the Medical Physics Department at Bradford Teaching Hospitals NHS Trust.

MULTIAGENT SYSTEM FOR FLOW MANAGEMENT IN COMPLEX SYSTEMS: DEVELOPMENT OF A DECISION-SUPPORT SYSTEM IN EPIDEMIOLOGY

ALEXANDRE WEBER[1, 2, *], DANIEL DUPONT[2,3], ANNE FOLLET[4], PHILIPPE KUBIAK[1], AHMED RAHMANI[1]

[1] LAGIS, UMR CNRS 8146, Ecole Centrale de Lille, Cité Scientifique - BP 48 59651 Villeneuve d'Ascq
[2]Lab. ERASM-HEI, Hautes Etudes d'Ingénieur 13, rue de Toul 59046 Lille Cedex
[3]CERNS, Chaire d'Enseignement et de Recherche Norbert Ségard, 3, rue Norbert Ségard 59046 Lille Cedex
[4]Laboratoire E&S, Faculté Libre des Sciences et Technologies de Lille, 41, rue du port 59046 Lille Cedex
ALEXANDRE.WEBER@EC-LILLE.FR

KEY WORDS:

Modelling, Complex Systems, Flow management, Multiagent Systems (MAS), Bioinformatics, Parasitology.

ABSTRACT:

Complex systems regroup a considerable number of entities. The number and diversity of these is determinant for the evolution of the system. This Evolution is, in particular, by stochastic relations which link the actors of the system. This evolution is not very predictable. Our work is centred on the flow circulation modelling and comprehension that takes place between the elements of the system. Taking account of the various specifications of complex systems, MultiAgent Systems (MAS) represent one of the most adapted methods for their simulation. It is possible to represent the emergence of environmental phenomena as the consequence of the interactions of agents acting in parallel. A particular flow management application on which we work is to simulate the circulation of a parasite (*Cryptosporidium* parvum) in an ecosystem. The objective is to better understand the various infection episodes of host populations (animal or human) under specific constraints. Given that this parasite is particularly resistant to the traditional disinfection methods, we want, in the long term, to produce an autonomous decision-support system in epidemiology which we have called Meta-MAS. The objective is to evaluate the factors acting in the parasite contamination and propagation.

STATE OF THE ART

An important property of complex systems is to regroup a considerable number of co-acting and interacting entities. This causes the internal organisation of the system to evolve in ever more complex ways (Coquillard and Hill 1997).

In fact, the multitude of interactions taking place between a great diversity of individuals makes the evolution of the system particularly difficult to predict (Bonabeau and Theraulaz 1994). To try to address this problem, the traditional analytical methods are based on two complementary points of view (Coquillard and Hill 1997). Although certain complex problems do not have an analytical solution (Le Moigne 1990), these two ways of proceeding have shown their reliability in many applications. However, they consider the systems in a partial and incomplete way by taking either a global or local approach (Coquillard and Hill 1997). The holistic point of view retains only the dynamic and thus ignores the subjacent system phenomena which are the result of many interactions. The reductionist point of view, for its part, divides the complexity of the system into distinct elementary components to represent the behaviour of only one entity which is isolated from the system (Coquillard and Hill 1997).

The Distributed Artificial Intelligence tools (DAI) propose to proceed to the modelling of the behaviour of each element which participates in the evolution of the system being studied. Their interactions with the other elements and their environment are also modelled. That allows a joint observation of the individuals (which are often of several different types) and population in evolution under specific environmental conditions (Dupont et al. 2002).

Consequently, it is no longer necessary to have an exhaustive knowledge of the system which means it does not have to be entirely characterised. The principle is to connect a significant number of autonomous entities having simple characteristics which will enable them to interact with each other in a specific environment (Dupont et al. 2002; Ferber 1995). At the time of these interactions, different standard data are exchanged: information, food, virus, etc… This data transfer can be considered as a flow. It is the modelling and the comprehension of this flow circulation in complex systems which is the subject of our study.

Multiagent systems (MAS) make use of this principle and, although suffering from a lack of tools for systematic analysis, seem like the ideal tool for this type of modelling. A MAS makes it possible to connect local causes (agent behaviour) with total results (general observation of group behaviour) and to reflect on how an individual behaviour can modify the collective one (Ferber 1994; Ferber 1995).

In fact, the MAS enable us to approach the reality of complex systems by recreating them and also allow us to act on the characteristics of individuals and the environment.

[*] The authors thank the NORBERT SEGARD Foundation for its support.

Our vision of the collective actions emerges from the direct or indirect interactions between the individuals and the environment. It is the result of a self-organisation process during which the environment and the community are mutually structured (Ferber 1995; Hill 1993). Since we model only the individual behaviour of agents and their interactions (figure 1), it is necessary to use a simulation in order to observe the system in its entirety.

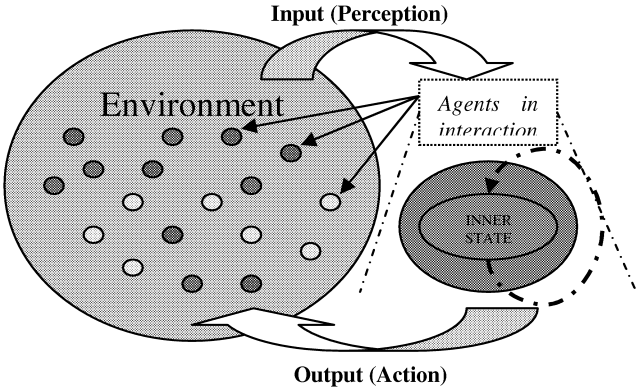


Figure 1: Working of a SMA

A methodical study using scenarios describing a precise environmental context is achieved using multiple data. This makes it possible to formulate various hypotheses or possible decisions and to observe their consequences (Le Moigne 1990; Hill 1993).

Simulation enables us to manage many parameters for each entity. This management of the individuals and of their physiological variations, allow the user to refine the model with the desirable degree of precision to be as close to reality as possible (Bonabeau and Theraulaz 1994; Ferber 1995) Modelling then arises from the observation of the results of the simulation (Hill 1993).

The study presented is within the framework of the modelling and the comprehension of flow circulation in complex systems. The exchanges which take place at the time of the interactions are comparable to a sequence of data which is transmitted from one point to another in a communication network i.e. to a flow.

This principle was applied to the parasites flow circulation between individuals in a natural ecosystem in order to evaluate the possible dissemination causes.

APPLICATION IN EPIDEMIOLOGY

Problem Specificities

As regards public health and the environment, the problems involved in water quality are at the forefront of concerns. Among the pathogenic agents circulating in water and posing considerable medical problems is an emergent kind: *Cryptosporidium*. This parasite can cause serious intestinal diseases in humans or domestic mammals (Derouin et al. 2002a; Derouin et al. 2002b).

The contamination is made in a feco-oral way, either in contact with infected individuals, or via contaminated food or water. The risk of infection is important for immunodeficient individuals (Derouin et al. 2002a; Derouin et al. 2002b).

In farm animals and cattle, the cryptosporidiosis is an important parasitosis. Indeed, *Cryptosporidium* is regarded as the major infectious agent in bovine, ovine and caprine neonatal diarrhoea syndrome, inducing considerable economic losses on livestock farms (Derouin et al. 2002a; Derouin et al. 2002b).

Today, molecular biology makes it possible to advance in the detection and identification of the various species of *Cryptosporidium*. However, it is also necessary to progress in the understanding of parasite circulation in ecosystems. For that, computer science plays a complementary role for the tracing of this pathogen in the environment. The modelling of the circulation of this parasite allows us to simulate cases of infection in populations present in a given ecosystem.

The interest of such modelling is to evaluate the influence of parameters on the general behaviour of a population of parasites at the ecosystem level. The objective is to design a computer tool starting from experimental biological data then making it possible to simulate the circulation of the parasite in a given environment and to study the ways in which it could infect a host population (animal or human). The primary finality of the project is the application of this modelling in epidemiology. This scientific approach can also be extended to the study of biotopes. Then, we take into account the variation of environmental factors influencing the parasite circulation and the possible establishment of the parasitosis.

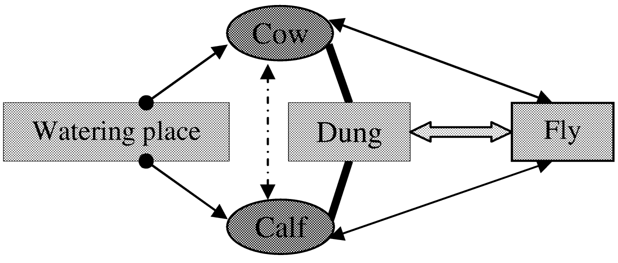
All the necessary data is provided by the team of biologists working with us on this study and by a bibliographical study (Derouin et al. 2002a; Derouin et al. 2002b). A data-processing tool for the simulation of the circulation of the parasite has been created. Initially, the ecosystem was restricted to a cattle shed (Bovines in boxes, human intervention, etc…).

Simulation Archetype

The computer tool, based on the principle of multiagent systems (MAS), makes it possible to define all the agents of the ecosystem and their individual characteristics. All the contacts between agents are also identified and modelled, in particular the number of exchanged parasites and their influence on the behaviour of concerned entities. The influence of the temperature and the humidity as well as the behavioural variations between the day and the night are also taken into account. Indeed, the evolution of the agents is strongly conditioned by these two parameters (fly reproduction, contacts between the agents…). In fact, the proliferation of the parasite is favoured during the wet seasons (Follet 2005), hence the need to consider these parameters.

All these processes are not fully deterministic, which is why, to simulate distribution laws, stochastic methods are called upon.

The difficulty of the model's conception lies in each entity's definition, interactions and in the way in which this affects each individual's behaviour (Ferber 1994; Ferber 1995). In order to establish the computer model, we had to define the various elements of this circulation. We established a diagram allowing the schematization of the circulation of *Cryptosporidium* in a cattle shed (Figure 2: Modelled SMA Representation)



Interactions between agents		Agents taken into Account in the SMA	
Contact	↔	Well / Tank	▭
Production	▬	Transformation agent	○
The dung/fly contact is symbolized differently because of reproduction of the flies in the dung.			

Figure 2: Modelled SMA Representation

Thus, "agents" and interactions in the form of "contacts" are defined. An "agent" represents a direct participant in this circulation and a "contact" represents the circulation route of the parasite between the actors. Two agent types are defined: the wells or tanks represent areas where the parasites are simply carried and the transformation agent symbolise areas where the parasites actively reproduce.

The bovines are characterised by their capacity to resist infection caused by the parasite. The ImmunoCompetence Status (ICS) represents this state by a value ranging between 0 and 1 (0, slightly immunocompetent, 1, strongly immunocompetent). With regards to the behaviour of the bovines with the aggressions of the parasite, we introduced an evolution law of the number of parasites according to the ICS and to the ingested quantity of *Cryptosporidium.* We will see hereafter that this law differs according to maturity from the immune system of the host. The other type of agents (tanks: dung, water, fly) are specified by the number of parasites that they convey. Certain authors think that the fly and the watering place are transformation agents. But, in the biological community, it is usually agreed that they must be regarded as "tanks" (Follet 2005; Graczyk 2000). Human action on the environment is also modelled in the simulation although this does not appear in the preceding diagram (cleaning of the cattle shed for example).

Agents and Evolution

The assumptions selected are founded on the literature and primarily on the data provided by the AFSSA document: "Quantitative Evaluation of the Medical Risk Related to the Presence of *Cryptosporidium* sp. in the water of distribution" (2002) (Derouin et al. 2002b). Because of a lack of quantified information relating to the bovines, all the data concerning the infecting amounts are established starting from studies on humans, the values being then transposed to the bovines. We will see hereafter the effects of this transposition.

The study considers two populations: an immunocompetent population and an immunodepressive one. Concerning the immunocompetent population, the data are obtained starting from the work of DuPont et al. 1995 and Okhuysen et al. 1999. As for the immunodepressive population, a transposition is made starting from the work completed in mice by Yang et al. 2000. Figure 3 represents the behaviour of the bovines (Derouin et al. 2002b) and is used as a working hypothesis for the realisation of the agents.

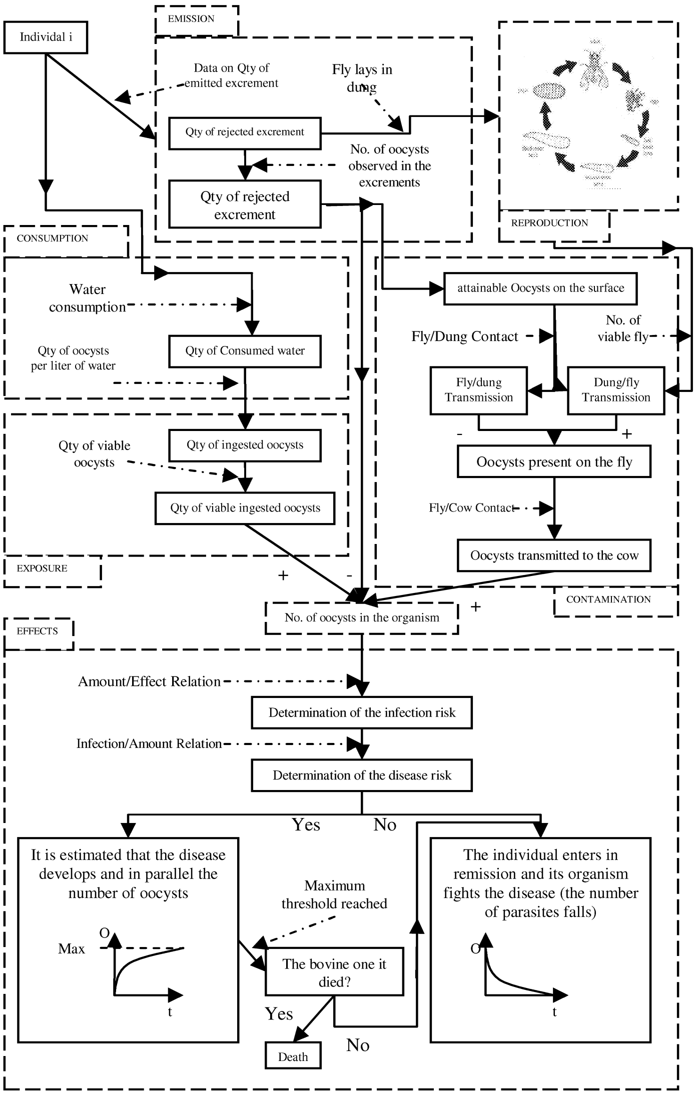

EMISSION
Individal i
Data on Qty of emitted excrement
Fly lays in dung
Qty of rejected excrement
No. of oocysts observed in the excrements
Qty of rejected excrement
REPRODUCTION
CONSUMPTION
Water consumption
Qty of oocysts per liter of water
Qty of Consumed water
attainable Oocysts on the surface
Fly/Dung Contact
No. of viable fly
Fly/dung Transmission
Dung/fly Transmission
-
+
Oocysts present on the fly
Qty of viable oocysts
Qty of ingested oocysts
Qty of viable ingested oocysts
Fly/Cow Contact
Oocysts transmitted to the cow
EXPOSURE
+
-
CONTAMINATION
No. of oocysts in the organism
+
EFFECTS
Amount/Effect Relation
Determination of the infection risk
Infection/Amount Relation
Determination of the disease risk
Yes
No
It is estimated that the disease develops and in parallel the number of oocysts
Max
O
t
Maximum threshold reached
The bovine one it died?
Yes
No
Death
The individual enters in remission and its organism fights the disease (the number of parasites falls)
O
t

Figure 3: Behaviour of the Bovines

Seasonal Variation and Influence on the Agents

In order to represent the diversity of the climate, we studied the variation in temperature and humidity according to time in five French towns as well as possible representations of the five climate types present in moderate zones. In order to constitute a model which is as close to reality as possible, we based our study on the daily data of MéteoFrance for several years in each of these cities.

We obtained a monthly model of temperature evolution starting from these data with the interpolation by splines method which is based on the Lagrange method[1] but avoids the oscillations caused by this with certain orders.

We have also to represent the variations between the day and the night since the data obtained were the minimal, maximal and average temperature of each day. For each day, we associate the temperature obtained with Gaussian in order to reach a random value.

To define each law in one precise moment of a day and to obtain a constant evolution of the temperature, some precise details must be brought to our model. Indeed, for the daytime we consider that the minimal temperature is with the rising and the setting of the sun with a maximal peak between 12 o'clock and one o'clock. For the night-time, it is the reverse; i.e. we consider the maximal temperature with the setting and the rising of the sun with a minimal peak between 2 o'clock and 3 o'clock in the morning. Then, we linearly distribute the temperatures during the day between the established values by preserving the stochastic aspect by the Gaussian associated ones. The linear temperature will be thus near the temperature defined by the linear curve of figure 4:

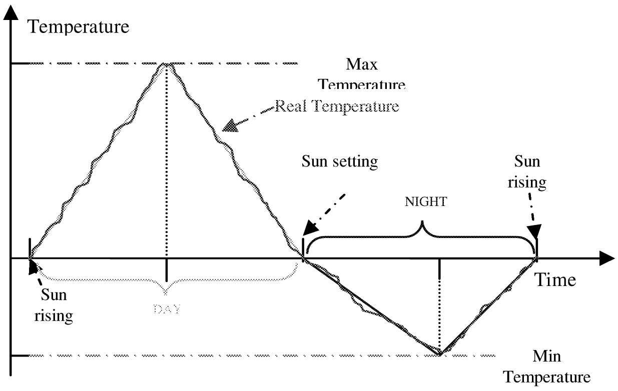


Figure 4: Evolution of the Temperature during the Day

With an aim of modelling the time of the rising and setting of the sun, we obtained from MéteoFrance these times for the 21st of each month for many years (because the solstices appear on the 21st of June and December). We realise these values to be able to carry out an interpolation by intervals to obtain two curves giving the times of the rising and the setting of the sun according to the date and to each climate type.

We used the same method with regard to humidity for each listed climate type.

These parameters influence the behaviour of the agents. The fly will move more or less according to the day/night variations, the watering frequency of bovines is modified according to the temperature as well as the contact frequency between the cow and the calf, deaths and births of the flies are entirely conditioned by these two variables. The evolution of the group strongly depends on this temperature and thus of the time of the year when we will choose to undertake an experiment.

Transformation Agent: the Bovines

These agents are defined by three points: their immunocompetence status (ICS), the number of oocysts (parasite infecting stage) carried and number of viable infectious oocysts carried. Several stages, intervening in the agent's behaviour (Figure 3: Behaviour of the bovines), make it possible to define and make these values evolve.

First of all, the bovines regularly consume a quantity of water governed by a binomial distribution and the evolution of the temperature. The parasite being conveyed by water, the user fixes an oocysts rate per litre of water. Thus, we determine a number of consumed oocysts (Oc) according to this contamination. The exposure evaluation of the bovines results from a combination of the quantity of the consumed and viable oocysts rising to 40% according to our data (Derouin et al. 2002b). Then, we estimate a number of viable oocysts (Ov) by comparing the proportion of 40% to a probability. Then, Ov follows a binomial distribution law of Oc parameters and 0.4:

$$O_V \sim Binomial(O_C, 0.4) \qquad (1)$$

The probability of an individual being infected is related to the number of ingested micro-organisms. Each oocyst in this amount (total number of ingested oocysts) has the same probability, noted r, to cause an infection. r is different from 1 because the parasite must survive the local defences of the organism and arrive on the infection site. Thus we define a law allowing a correlation between infection and the amount:

$$\Pr(Infection / Amount) = 1 - e^{-r \times Amount} \qquad (2)$$

In order to define the probability r, we introduce the concept of "Infectious Amount 50" (IA50). It represents the quantity of parasites causing the infection of half of a herd of contaminated bovines. The IA50 corresponds to the amount such as:

$$\Pr(Infection / Amount) = 0.5 = 1 - e^{-r \times IA50} \qquad (3)$$

We deduce:

$$r = \frac{-\ln(0.5)}{IA50} \qquad (4)$$

[1] Lagrange Interpolation: Distinct points x0, x1,…., xn of a limited closed interval [a, b]єℝ, and a function f defined on [a, b] in values in ℝ, there is a single polynomial P such as P(xi) = F(xi) for i = 0, 1,…., n. This polynomial is given by:

$$P(X) = \sum_{i=0}^{n} \left(f(x_i) \prod_{\substack{j=0 \\ j \neq i}}^{n} \frac{X - x_j}{x_i - x_j} \right)$$

The IA50 is equal to 165 oocysts for an individual resulting from an immunocompetent population and is established with 1,96 oocysts for an individual resulting from an immunidepressed population (Derouin et al. 2002b; DuPont et al. 1994; Okhuysen and Al 1999). Thus, we have:
r = 0.00419 for an immunocompetent population (ICS = 1)
r = 0.35365 for an immunodepressed population (ICS = 0)
A linear extrapolation enables us to connect the probability r to the ICS. As follows:

$$r = -0.34946 \times ICS + 0.35365 \qquad (5)$$

In reference to figure 3, we notice that it is not because an individual ingests an infectious amount that it develops the cryptosporidiosis. The probability of developing the disease while being a carrier is independent of the ingested amount. It amounts to 39% for an immunocompetent population and 100% for an immunodepressed population (Derouin et al. 2002b). Variable ICS selected lying between 0 and 1, by linear extrapolation, the established relation is:

$$\Pr(Disease\,/\,Infection) = p = -0.61 \times ICS + 1 \qquad (6)$$

Once ingested, we are interested in the evolution of the parasite within a cow (calculations are the same for a calf). It is considered that the cow can carry a maximum of 10^8 oocysts (10^7 oocysts for a calf). We calculate randomly a maximum threshold which is specific to each individual (MaxCowCrypto). This threshold, of which we saw the interest in figure 3, depends on the bovine health (ICS).

$$MaxCowCrypto = Max(\frac{10^8}{1000}, Binomial(10^8, 1 - ICS)) \; (7)$$

We secure a minimum threshold of portable parasites of 10^5 and to obtain a random value around ICS * 10^8. We should then determine the number of infectious oocysts (Oi). However, according to the equation (2), we have:

$$Amount = \frac{-\ln(1 - \Pr(Infection\,/\,Amount))}{r} \qquad (2bis)$$

We can thus calculate an amount according to the probability p that this amount has to cause an infection. We take again this law in order to withdraw an amount of non-infectious oocysts from the quantity of viable oocysts (Ov) calculated by:

$$O_v = Max(\frac{qc}{100}, Binomial(qc, 0.4)) \qquad (1bis)$$

As in (1), there is a random quantity of oocysts in the neighbourhood of 40% of the ingested quantity and we limit calculation to 1% of this quantity. We calculate then the quantity of infectious oocysts (Oi) in the following way:

$$O_i = O_v - \frac{-\ln(p)}{r} = O_v + \frac{\ln(p)}{r} \qquad p \in [0,1] \text{ random} \qquad (8)$$

As r is connected to the ICS (cf (5)), the larger the ICS is the more likely we will be to withdraw a significant number of oocysts from the initial amount. For a given ICS, the larger the probability p, the smaller the withdrawn number of parasites and thus the larger the Oi is.

Due to the quantity of ingested oocysts, the individual can develop the Cryptosporidiosis. If it develops it the Oi amount can be lethal. Various rules were defined. The first determines if the cow develops the Cryptosporidiosis. That is to say p, a random value, if p is lower than the probability of developing the disease for a given ICS (cf (6)) and that the individual has never been sick, then the disease breaks out. The ingested amount is defined as lethal according to the ICS: the weaker the ICS is, the greater the probability that the amount is lethal.

Once the amount was ingested, the quantity of oocysts follows an evolution law (of exponential type) dependant on the ICS and on the development of the Cryptosporidiosis. A sick cow can recover if it is supposed that the introduced amount is not lethal. In this case, if it reaches the maximum number of oocysts, we consider that it enters in remission. On the other hand, a cow dies if the introduced amount is lethal and it reaches the maximum number of oocysts.

The « Tank » Agents

The “flies” are regarded as “tank” agents because they are passive conveyors (Follet 2005; Graczyk 2000). This agent is defined by the maximum number of oocysts which it can carry and by the number of viable infectious oocysts carried. The maximum number of oocysts which a fly can carry is different for each individual and is defined by:

$$MaxCowCrypto = Max(\frac{10^3}{1000}, Binomial(10^3, 0.8)) \qquad (9)$$

The flies collect parasites in the environment according to the various interactions in which they take part. Just as previously, we determine randomly the number of infectious oocysts (Oi) among the total quantity of collected parasites (qc). Of course, we respect the proportions given by the biologists i.e. 10% at least and 80% on average of infectious oocysts per insect after contact with a parasitized substratum:

$$O_i = Max(\frac{qc}{10}, Binomial(qc, 0.8)) \qquad (10)$$

The “dung” is defined by the number of accessible oocysts on its surface (Oa) and by the number of viable infectious oocysts carried (Ov). When dung is produced, at least 20% and on average 90% of the oocysts contained in the dung (qc) are infectious. The oocysts contained in dung are not all accessible, only those on surfaces are. It is considered that at least 1% and on average 10% is accessible:

$$O_v = Max(\frac{qc}{5}, Binomial(qc, 0.9)) \qquad (11)$$

$$O_a = Max(\frac{O_v}{100}, Binomial(O_v, 0.1)) \qquad (12)$$

The “watering place” is defined by the number of viable infectious oocysts per litre (Ov). It is considered that on average 40% of the oocysts found in water are viable with a minimum of 5% (Derouin et al. 2002b). Moreover, the viable oocysts are all regarded as potentially infecting (weak[2] and safety[3] hypothesis).

$$O_v = Max(\frac{qc}{20}, Binomial(qc, 0.4)) \qquad (13)$$

Modelling of Interactions

Each interaction proceeds between two agents of the environment. These interactions are comparable with contacts at the time during which a flow of parasites is exchanged between the concerned agents. We determine for each contact a transmitter and a receiver of oocysts. Several scenarios are possible depending on the actors of the contact: the bovines drink (contacts Water → Bovine), flies land on the muzzle of the bovines, flies land on dung (the fly contaminates the dung or conversely), bovines defecate (contacts Bovine → Dung) and the calves suck or the cows lick them (contacts Bovine→ Bovine)

Emission of the Watering Place
The information, received from the biologists, enabled us to know the various data which we needed on the watering of the bovines. They are resumed in the table below:

Table 1: Data on Bovine Watering

Agents	Minimal quantity of absorbed water by catch	Maximal quantity of absorbed water by catch	Average quantity of absorbed water by catch	Minimal quantity of oocysts ingested by catch	Average quantity of oocysts ingested by catch
Cow	2 litres	20 litres	10 litres	20%	80%
Calf	1 litre	10 litres	5 litres	25%	90%

For each interaction, the emission procedure determines successively the water quantity which is absorbed by the bovines (Q_drink), the number of transmissible oocyst in this water quantity (Q_transmissible) and the number of oocysts really transmitted by water, i.e. ingested by the recipient agent (Q_transmit). We represent by Ov the water contamination in number of parasites per litre. According to data stated above, we have:

Table 2: Calculation of the Quantity of Transmitted Oocysts at a Watering Time.

Contact	Q_drink	Q_transmissible	Q_transmit
Water → Cow	Max(2, binomial(20, 0.5))	Q_drink*Ov	Max(Q_transmissible/ 5,binomial (Q_transmissible,0.8))
Water → Calf	Max(1, binomial(10, 0.5))	Q_drink*Ov	Max(Q_transmissible/ 4,binomial (Q_transmissible,0.9))

Emission of the fly
The fly has three possibilities to transmit the parasites which it conveys (Ov). It lands on the muzzles of cows or calves or on dung and contaminates them. For each one of these interactions, we suppose that at least 10% of the oocysts are transmitted and on average 50%:

$$O\_transmit = Max(\frac{O_v}{10}, Binomial(O_v, 0.5)) \qquad (15)$$

We then subtract the number of transmitted oocysts from the number of oocysts present on the fly.

Emission of the Bovines
The bovines emit parasites into the environment through their dejections because *Cryptosporidium* reproduces in the digestive system. Then the quantity of oocysts depends on the weight of defecated dung which we will note hereafter as "dung_weight" calculated in grams of faecal matter (FM).

Table 3: Data on Faecal Matter of the Bovines.

Agents	Minimal Qty of FM by dung	Maximal Qty of FM by dung	Average Qty of FM by dung	Minimal quantity of oocysts ejected by dung	Average quantity of oocysts ejected by dung
Cow	200g	700g	350g	5%	30%
Calf	100g	300g	175g	5%	30%

From these data we define the following table:

Table 4: Calculation of the Quantity of Transmitted Oocysts When the Bovines Defecate.

Contact	dung_weight	Q_transmissible	Q_transmit
Cow → Dung	Max(200, binomial(700, 0.5))	O_v* dung_weight /1000	Max(O_transmissible /20,binomial(O_transmissible, 0.3))
Calf → Dung	Max(100, binomial(350, 0.5))	O_v* dung_weight /500	Max(O_transmissible /20,binomial(O_transmissible, 0.3))

As for the fly, we subtract the transmitted quantity from the number of occysts present in the transmitter.

Dung Emission
The dung can transmit oocysts to the fly which lands there. It is thus necessary to calculate the number of attainable parasites on the surface (Oat) for each dung. It is estimated that at least 1% and on average 20% of these transmitted oocysts by the bovine (Ot) are attainable on the surface:

$$O_{at} = Max(\frac{O_t}{100}, Binomial(O_t, 0.2)) \qquad (16)$$

The dung can transmit part of these attainable parasites to the fly (qc). It is estimated that the fly at least collects 5% and on average 50% of the attainable oocysts on the dung. Then the following calculation is obtained:

$$qc = Max(\frac{O_{at}}{20}, Binomial(O_{at}, 0.5)) \qquad (17)$$

SIMULATIONS RESULTS

Scenarios Establishment

We placed in the cattle shed 5 cows, 5 calves and 5 flies; cows and calves having each one an ICS growing (of 10 into 10: from 60 to 100 for cows and from 20 to 60 for calves). Cattle shed cleaning once per day is symbolised by

[2] A hypothesis is known as *safety* if its adoption tends to over-estimate the risk.
[3] A hypothesis is known as *strong* if the authors consider that this assumption is not realistic; it is known as *weak* if not.

the elimination of the oocysts in the environment. Lastly, we carry out simulations over approximately ten day periods.

We set up six types of scenarios in order to give a first series of brief replies to the biologists and also with the aim of testing the precision of the software. Each scenario is carried out 20 times.

Scenario A, relates to the arrival of oocysts only via water by the distribution network at a rate of 5 oocysts per litre in continuous flow. In scenario B, the arrival of oocysts is done by the flies. We enter 10 oocysts per fly so there are a total of 50 oocysts in the cattle shed at the start of the simulation. Scenario C relates to the arrival of oocysts by the cows: each cow carries 10 oocysts which makes a total of 50 oocysts. The three following scenarios correspond to the arrival of oocysts by calves. With scenario D, it is a 6th calf from outside the cattle shed which brings 50 oocysts; we vary the ICS of this 6th calf from 0,2 to 0,6. Scenario E takes into account the arrival of oocysts by a calf from the cattle shed having an ICS of 0,2. As for scenario F, it consists in varying the quantity of oocysts carried by a calf having an ICS of 0,2. We test 25, 50, 75, 100, 500 and 1000 oocysts.

Results and Contributions of the Scenarios

The six scenarios previously quoted provide different information in relation to the role of the immunocompetence status (ICS) of the vector and of the host and in relation to the number of oocysts entering the system.

The results obtained show that the livestock can be infected by *Cryptosporidium*, that the oocysts are conveyed by water or by flies (scenario A, B). Indeed, as we can see in Figure 5, the whole population was infected. Moreover, the calves 1 and 2, which have respectively an ICS of 0,2 and 0,3, died most of the time (as the red cross indicates on the graphs). The number of oocysts decreases progressively with the ICS level for the calves. For the cow, we can see that the number of oocysts is low and about the same for the whole population.

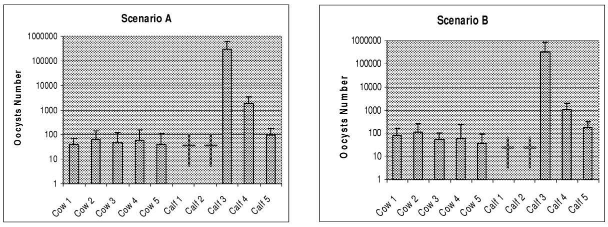


Figure 5: Logarithmic Graphs of Scenario A and B Average Results

So the adult cows, whatever their ICS is, always present a restricted quantity of parasites, without reaching a sufficient quantity to result in death. On the other hand, in the calves whose ICS is lower than 0,4, the parasites develop until reaching a mortal amount. The host ICS as well as the immune system maturation seems to play a role in disease development.

When the parasite vector is represented by an adult cow with an ICS higher than 0,6 (scenario C), the restricted quantity of parasites observed in the adult bovines is non-existent. This would suggest an elimination of the oocysts or a parasitic viability loss due to the immune system of the cow which carries them. In the same way, the number of oocysts found in calves is very low even null. The dissemination by the flies here is largely reduced since accessibility with the oocysts in the faecal matter is low even null.

From the preceding scenarios, it is highlighted that the ICS of the vector plays a considerable role. In order to confirm this observation, scenario D was elaborated by using a 6th carrier calf. If the initial calf, carrier of the parasite, presents an ICS higher than 0,4, then the parasitic development in the ecosystem no longer presents sufficient numbers of parasites to result in death, even for the herd calves having a weak ICS (Figure 6). Thus, this would confirm that the animal ICS in which the parasite develops is an important factor in the dissemination of the parasite.

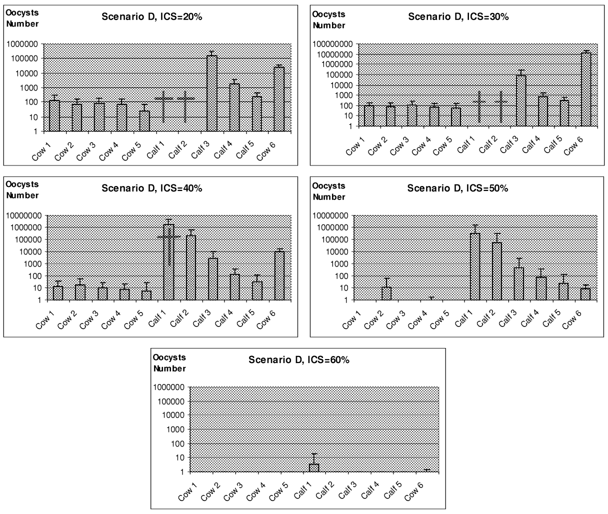


Figure 6: Logarithmic Graphs of Scenario D Average Results

Scenario E shows that in spite of the bovines' capacity to allow the multiplication of the parasite, the amplification of the number of oocysts does not influence the final destiny of the herd. Indeed, if the vector is a calf having an ICS of 0,2, the result is then the same as with an introduction of oocysts by water or flies (scenario A and B).

The last evaluated scenario (scenario F) aims at determining the importance of the number of parasites present in the initial carrier calf (with an ICS of 0,2) on the risk of cryptosporidiosis in the herd. The results obtained confirm the assumption suggested in scenario D. Thus it would not be the number of parasites which arrives in the system which is important, but the host ICS at which it develops.

CONCLUSION AND PERPECTIVES

For a few years, the evaluation of the risk related to the presence of *Cryptosporidium* in the environment has represented an increasing pole of interest. Authors like Warbler et al. (Warbler et al. 2004) developed a method for the evaluation of risks concerning the French human population. The computer modelling which was accomplished in this study relates more to an epidemic problem in a breeding circle, but it could be adapted to problems concerning human health.

In order to build our model, many working hypotheses had to be posed and remain debatable. An evaluation of the risks was taken into account. All the hypotheses were postulated starting from the literature and expert knowledge (Derouin et al. 2002a; Derouin et al. 2002b).

With the results obtained and their analysis by the biologists as well as the comparison with the bibliographical data, the model seems to faithfully reproduce the circulation of *Cryptosporidium* in the restricted environment of the cattle shed. The qualitative results concerning the role of immunocompetence status (ICS) compared to the parasitic development corroborate the bibliographical data (Subdued et al. 1984; Holten-Andersen et al. 1984). In this study, the role of dipterous in the dissemination of the parasite is also highlighted. These observations reinforce the assumption of passive transport of the oocysts already under consideration by Graczyk et al. (Graczyk et al. 2001). In addition, the scenarios enabled us to test the robustness and sensitivity of the software. Indeed, we showed the emergence of recurring group phenomena like the ICQ threshold appearing in several scenarios. The appearance of isolated events due to the variation of the initial parameters also could be observed. However, according to biologists, these events correspond in their frequency and their form to the random element which we find in nature.

The evolution which we propose comes as an addition to this first basic model like a system of a higher nature able to supervise it. This meta-system, called Meta-MAS, retroacts on the initial data by proposing new scenarios. The finality is to extract the causes from emergent events while retroacting on the characteristics of the individuals and the environment. So, the Meta-MAS modifies the evolution of the system and is able to observe and quantify these modifications.

It makes hypotheses on the origin of particular flow propagation. Its hypotheses were checked and the model reaction is again tested in order to evaluate the repercussion of a change of parameters. This will make it possible to increase the knowledge we have on the parameters and the role of their interactions. This knowledge enables us to determine on which elements it is necessary to act in order to limit (case of pandemias) or to increase flow.

In many complex systems (logistics, car traffic, economy…), it is possible to identify particular elements which increase or decrease the flow propagation. The combination of these elements can sometimes have unexpected effects like increasing or decreasing flow where separately they have the opposite effect. The Meta-MAS identifies these elements to index them and then test their influence on the evolution of the system. Thus, the Meta-MAS is able to find the best configuration of joint elements which optimises the system.

REFERENCES

Bonabeau E., Theraulaz G., 1994. “Intelligence Collective”, Hermes.

Coquillard P., Hill D., 1997. “Modélisation et Simulation d’Ecosystèmes: des Modèles Déterministes aux Simulations à Evénements Discrets”, MASSON.

Derouin F., Beaudeau P., Pouillot R., Roze S., 2002. “Evaluation Quantitative du Risque Sanitaire lié à la Présence de *Cryptosporidium sp*. dans l’Eau Distribuée”, AFSSA.

Derouin F., Eliaszewicz M., Pouillot R., Roze S., 2002. “Rapport sur les Infections à Protozoaires liées aux Aliments et à l’Eau : Evaluation Scientifique des Risques associés à *Cryptosporidium* sp.”, AFSSA.

Dupont D., Kökösy A., Biela P., Saadane A., 2002. “Vie artificielle: application à la résolution de problèmes complexes”, Techniques de l’ingénieur.

DuPont H.L., Chappell C.L., Sterling C.R., Okhuysen P.C., Rose J.B., Jakubowski W., 1995. “The Infectivity of *Cryptosporidium parvum* in Healthy Volunteers.” *N Engl J Med*, *332, 855-859.*

Ferber J., 1994. “Coopération Réactive et Emergence”, Intellectica, 1994/2,19,pp. 19-52.

Ferber J., 1995. “Les Systèmes Multi-agents: vers une Intelligence Collective”, InterEdition.

Follet A., 2005. “Mise en Evidence du Rôle des Insectes (*Dipterae*) dans le Transport de *Cryptosporidium parvum”*, Thèse pour l'obtention du Doctorat des Sciences de l'Université de Lille II *Discipline: Parasitologie.*

Graczyk T.K., Fayer R., Knight R., Mhangami-Ruwende B., Trout J.M., Da Silva A.J., Pieniazek N.J.,2000. “Mechanical Transport and Transmission of *Cryptosporidium parvum* Oocysts by Wild Filth Flies”. *Am J Trop Med Hyg*, *63, 178-183.*

Graczyk T.K., Knight R., Gilman R.H., Cranfield M.R., 2001; “The Role of Non-biting Flies in the Epidemiology of Human Infectious Diseases.” *Microbes Infect*, *3, 231-235.*

Hill D., 1993. “Analyse Orientée Objet et Modélisation par Simulation”, Addison-Wesley France.

Holten-Andersen W., Gerstoft J., Henriksen S.A., Pedersen N.S., 1984. “Prevalence of *Cryptosporidium* among Patients with Acute Enteric Infection.” *J Infect*, *9, 277-282.*

Le Moigne J. L., 1990 “La Modélisation de Systèmes Complexes”, AFCET Systèmes, Dunod.

Mata L., Bolanos H., Pizarro D., Vives M., 1984. “Cryptosporidiosis in Children from some Highland Costa Rican Rural and Urban Areas.” *Am J Trop Med Hyg*, *33, 24-29.*

Okhuysen P.C., Chappell C.L., Crabb J.H., Sterling C.R., DuPont H.L., 1999. “Virulence of Three Distinct *Cryptosporidium parvum* Isolates for Healthy Adults.” *J Infect Dis*, *180, 1275-1281.*

Pouillot R., Beaudeau P., Denis J.B., Derouin F., 2004. “A Quantitative Risk Assessment of Waterborne Cryptosporidiosis in France using Second-order Monte Carlo Simulation.” *Risk Anal*, *24, 1-17.*

Yang S., Benson S.K., Du C., Healey M.C., 2000. “Infection of Immunosuppressed C57BL/6N Adult Mice with a Single Oocyst of *Cryptosporidium parvum*.” *J Parasitol*, *86, 884-887.*

APPROXIMATION AND EVALUATION SIMULATION

RELIABILITY-BASED PARETO OPTIMUM DESIGN OF ROBUST COMPENSATORS FOR A DYNAMIC SYSTEM WITH PARAMETRIC UNCERTAINTY

Nader Nariman-zadeh
Amir Hajiloo
Ali Jamali
Ahmad Bagheri
Faculty of Engineering
Department of Mechanical Engineering,
The University of Guilan, P.O. Box 3756, Rasht, IRAN
E-mail: nnzadeh@guilan.ac.ir

Aria Alasti
Faculty of Mechanical Engineering
Sharif University, Tehran, IRAN
E-mail: aalasti@sharif.edu

KEYWORDS
Uncertain systems, Robust compensator, Reliability, Multi-objective optimization, Genetic algorithm, Pareto.

ABSTRACT

A reliability-based approach for the Pareto optimum design of robust compensators for a dynamic system with probabilistic uncertainty is presented. In this way, some non-dominated optimum robust compensators in the Pareto sense are found using four non-commensurable objective functions both in time and frequency domains based on stochastic behavior of a system with parametric uncertainties. It is shown that multi-objective Pareto optimization of such robust compensators using a recently developed diversity preserving mechanism genetic algorithm unveils some very important and informative trade-offs among these objective functions.

INTRODUCTION

Synthesis of control policies can be presented as optimization problems of certain performance measures of the controlled systems. A very effective means of solving such optimum controller design problems is genetic algorithms (GAs) and other evolutionary algorithms (EAs). Some early applications of GAs in optimum design of controllers are reported in (Porter and Jones 1992) (Goldberg 1989) (Porter et al. 1994). In addition to the most applications of EAs in the design of controllers for certain systems, there are also much research efforts in robust design of controllers for uncertain systems in which both structured or unstructured uncertainties may exist (Wolovich 1994). Indeed, designing robust control method for uncertain systems is a computationally complex problem. Recently, there have been many efforts for designing robust control methods. In these robust design methods, probabilistic uncertainty propagates through the uncertain parameter of plants. The notions of stochastic robustness and probabilistic analysis have been first presented by Stengel (Stengel 1986) and Stengel and Ryan (Stengel and Ryan 1989). The analysis of Monte Carlo Simulation (MCS) has also been first introduced by Stengel to evaluate stochastic stability and performance of probabilistic uncertain systems (Stengel and Ryan 1989). GAs have also been recently deployed in an augmented scalar single objective optimization to minimize the probabilities of unsatisfactory stability and performance estimated by Monte Carlo Simulation (Wang and Stengel 2002). However, choosing appropriate weighting factors in a cost function consisting of weighted quadratic sum of those non-commensurable objectives is inherently difficult and could be regarded as a subjective design concept. Moreover, trade-offs existed between some objectives cannot be derived and, therefore, it would be impossible to choose an appropriate optimum design reflecting the compromise of the designer's choice concerning the absolute values of objective functions.

In this paper, a multi-objective genetic algorithm with a new diversity preserving mechanism recently reported by some of authors (Nariman-zadeh et al. 2005) (Atashkari et al. 2005) is used in conjunction with MCS to obtain Pareto frontiers of various non-commensurable objective functions in the design of robust compensators for an uncertain single-input single-output benchmark control problem.

STOCHASTIC ROBUST ANALYSIS

In real control engineering practice, there exist a variety of typical sources of uncertainties which have to be compensated through a robust control design approach. Two categorical types of uncertainty, namely structured uncertainty and unstructured uncertainty can be used in classification. The structured uncertainty concerns about the model uncertainty due to unknown values of parameters in a known structure. In conventional optimum control system design, uncertainties are not addressed and the optimization process is accomplished deterministically. In fact, it has been shown that optimization without considering uncertainty generally leads to non-optimal and potentially high risk solution

(Lim et al. 2005). Therefore, it is very desirable to find robust design whose performance variation in the presence of uncertainties is not high. Generally, there are two approaches addressing the stochastic robustness issue, namely, robust design optimization (RDO) and reliability-based design optimization (RBDO), (Papadrakakis et al. 2004). Both approaches represent non-deterministic optimization formulations in which the probabilistic uncertainty is incorporated into the stochastic optimal design process. Therefore, the propagation of *a priori* knowledge regarding the uncertain parameters through the system provides some probabilistic metrics such as random variables (e.g., settling time, maximum overshoot, closed loop poles, …), and random processes (e.g., step response, Bode or Nyquist plot, …) in a control system design (Smith et al. 2005). In RDO approach, the stochastic performance is required to be less sensitive to the random variation induced in uncertain parameters so that the performance degradation from ideal deterministic behavior is minimized. In RBDO approach, some evaluated reliability metrics subjected to probabilistic constraints are satisfied so that the violation of design requirements is minimized. In this case, limit state functions are required to define the failure of the control system.

Let *X* be a random variable, then the prevailing model for uncertainties in stochastic randomness is the probability density function (PDF), $f_X(x)$ or equivalently the cumulative distribution function (CDF), $F_X(x)$, where the subscript *X* refers to the random variable. This can be given by

$$F_X(x) = \Pr(X \leq x) = \int_{-\infty}^{x} f_X(x)dx \quad , \qquad (1)$$

where, $\Pr(.)$ is the probability that an event ($X \leq x$) will occur.

In the reliability-based design, it is required to define reliability-based metrics via some inequality constraints. It is now desirable to design a set of design parameters ($\mathbf{d}$) whose PDF can be given by $f_p(\mathbf{d})$ (or equivalently the CDF by $F_p(\mathbf{d})$) so that the reliability requirements given as

$$P_f(\mathbf{d}) = \Pr(g\ (\mathbf{d}) \leq 0) = \varepsilon \quad , \qquad (2)$$

is satisfied. In Equation (2), P_f denotes the probability of failure (i.e., $g(\mathbf{d}) \leq 0$) and ε is the highest value of desired admissible probability of failure. It is clear that the desirable value of P_f is zero. This integral is, in fact, very complicated particularly for systems with complex $g(\mathbf{d})$ (Wang and Stengel 2002) and Monte Carlo simulation is alternatively used to approximate Equation (2). Monte Carlo simulation (MCS) is a direct and simple numerical method but can be computationally expensive. In this method, random samples are generated assuming pre-defined statistical distributions for uncertain parameters. The system is then simulated with each of these randomly generated samples and the percentage of cases produced in failure region defined by limit state function approximately reflects the probability of failure. In this case, a binary indicator function $I_{g(x)}$ is defined such that it has the value of 1 in the case of failure ($g(\mathbf{d}) \leq 0$) and the value of zero otherwise,

$$I_{g(\mathbf{d})} = \begin{cases} 0 & g(\mathbf{d}) > 0 \\ 1 & g(\mathbf{d}) \leq 0 \end{cases} \quad . \qquad (3)$$

Based on Monte Carlo simulation (Wang and Stengel 2002) the integral of Equation (2) using sampling technique can be estimated using

$$P_f(\mathbf{d}) = \frac{1}{N}\sum_{i=1}^{N} I_{g(\mathbf{d})}(G(\mathbf{d}), C(\mathbf{k})) \quad . \qquad (4)$$

where, $G(\mathbf{d})$ is the uncertain plant model and $C(\mathbf{k})$ is the controller to be designed in the case of control system design problems and *N* is the total number of samples. In other words, the probability of failure is equal to the number of samples in the failure region divided by the total number of samples. Evidently, such estimation of P_f approaches to the actual value in the limit as $N \to \infty$ (Wang and Stengel 2002). However, there have been many research activities on sampling techniques to reduce the number of samples keeping a high level of accuracy. Alternatively, the quasi-MCS has now been increasingly accepted as a better sampling technique witch is also known as Hammersley Sequence Sampling (HSS) (Smith et al. 2005). In this paper, HSS has been used to generate samples for probability estimation of failures. In a RBDO problem, the probability of representing the reliability-based metrics given by Equation (4) is minimized using an optimization method. Moreover, in a RDO problem the upper bound of a random variable is minimized using an optimization method. Figure (1) depicts the concept of RDO approach adopted in this paper where $f_x(x)$ is PDF of random variable, *X*.

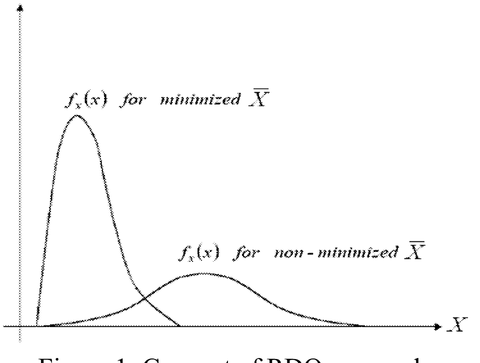


Figure 1: Concept of RDO approach.

It is clear form Figure (1) that if the upper bound of *X* is minimized, a robust optimum design can be obtained. The goal of this approach is to minimize the mean of the random variable as well as its variance (Kang 2005).

In the multi-objective mixed RDO and RBDO of control system problems, such robust metrics and reliability-based metrics (objective functions) can be selected as closed-loop system stability, step response in time domain, control effort, etc. In the probabilistic approach, it is, therefore, desired to minimize both the probability of instability and probability of failure to a desired time response, respectively, subjected to assumed probability distribution of uncertain parameters. Recently, a weighted-sum multi-objective approach has been applied to aggregate these objectives into a scalar single-objective optimization problem (Wang and Stengel 2002) (Kang 2005). However, the trade-offs of the objectives are not revealed unless a Pareto approach of optimization is applied.

PROBLEM DESCRIPTION

The benchmark plant that is used in this paper is a two-mass-spring system depicted in Figure (2) in which x_1 and x_2 represent the positions of the masses m_1 and m_2, respectively.

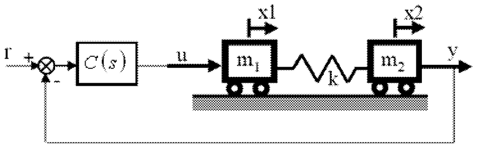


Figure 2: Two mass spring system.

The output of the controller *u* acts on m_1 to control the position of mass m_2. The transfer function between position of m_2, y, and control effort, *u* , is given as

$$G(s)=\frac{\dfrac{k}{(m_1m_2)}}{s^2\left[s^2+k\left(\dfrac{m_1+m_2}{m_1m_2}\right)\right]} \quad . \tag{5}$$

There are many research efforts that were accomplished using different techniques to produce the compensator for this system (Smith et al. 2005). The nominal values of three parameters of system are $m_1 = m_2 = 1$, $k = 1$ each with a beta distribution having shaping coefficients 2 and 2 in a range of $\pm 50\%$ of the nominal values. In this paper, the following compensator structure is used

$$C(\mathbf{k})=\frac{k_1+k_2s+k_3s^2+k_4s^3}{k_5+k_6s+k_7s^2+k_8s^3} \quad , \tag{6}$$

where the design vector $\mathbf{k}=[k_1,k_2,\cdots,k_8]$ has to be optimally determined based on the mixed robust and reliability-based multi-objective Pareto approach for the uncertain first-order system using some stochastic evaluation metrics.

The performance of a controlled closed-loop system is usually evaluated by a variety of goals. The most important loop goal is the robust stability which implies that all the closed-loop poles of the system remain in the stable left half-plane ($\Re(s_i)<0$) in the presence of any uncertainty for the nominal plant's transfer function $G(s)$. The probability of failure of stochastic stability can be computed using Equation (4)

$$\Pr_{ins}=\frac{1}{N}\sum I_{g_{ins}(\mathbf{d})}(G(\mathbf{d}),C(\mathbf{k})) \quad , \tag{7}$$

in association with Equation (7) employing the quasi Monte Carlo Sampling or HSS for *N* samples.

A good time-response performance is another measure whose probability of failure should also be minimized through the quasi Monte Carlo simulation. The lower and upper failure boundaries to define the corresponding limit state function, $g_{resp}(\mathbf{d})\le 0$, is given using the Heaviside function

$$\underline{y}=0.8\mathrm{H}(t-5)+0.15\mathrm{H}(t-10) \tag{8-a}$$

$$\overline{y}=1.2\mathrm{H}(t)-0.15\mathrm{H}(t-10) \tag{8-b}$$

for a period of $t\in[0,15]$. The probability of failure to the desired time response boundaries is then obtained using Equation (4) and that the time response y must resides between $\underline{y}$ and $\overline{y}$, that is $\underline{y}\le y\le\overline{y}$. Similarly, maximum amplitude of control effort, $\overline{u}$, is an important performance criterion should also be minimized.

In addition to the minimizing the probability of instability given by Equation (7), maximizing the stability margin in frequency domain is another important measure of good performance of a robust controller for uncertain systems. Such robust stability margin, also referred to as degree of stability, S_∞^{-1}, can be simply computed using the sensitivity transfer function

$$S_\infty^{-1}=\|S\|_\infty^{-1}=\min_\omega|1+L(j\omega)| \quad , \tag{9}$$

simply represents the minimum distance from -1 to $L(j\omega)$ in the Nyquist plot (Doyle 1992).

ILLUSTRATIVE EXAMPLE

The objectives $\Pr_{ins}, \Pr_{resp}$, $\overline{u}$, and S_∞^{-1} are now considered simultaneously in a Pareto optimization process to obtain some important trade-offs among the conflicting objectives. The vector of objective functions to be optimized in a Pareto sense is given as follows

$$\overrightarrow{cf}=fcn\left[\Pr_{ins},\Pr_{resp},\overline{u},S_\infty^{-1}\right] \quad . \tag{10}$$

The first three objectives need to be minimized while the fourth one, the degree of stability, has to be maximized. The evolutionary process of the Pareto multi-objective optimization is accomplished by using the modified NSGA-II approach presented by some of authors in (Nariman-zadeh et al. 2005) (Atashkari et al. 2005) where a population size of 80 has been chosen with crossover probability P_c and mutation probability P_m as 0.85 and 0.09, respectively.

The optimization process of the robust compensator given by Equation (6) is accomplished by 500 Monte Carlo evaluations using HSS distribution for each candidate control law during the evolutionary process. The vector of objective functions given by Equation (10) is used to obtain non-dominated optimum robust compensators to represent the trade-offs among the objective functions.

A total number of 62 non-dominated optimum design points have been obtained and shown in Figure (3) in the plane of probability of failure to the desired time response (Pr_{resp}) and the degree of stability (S_∞^{-1}). The value of probability of instability (Pr_{ins}) of all the non-dominated optimum points has been obtained zero which demonstrates that all optimum controllers are stable in the Monte Carlo simulation. Evidently, it can be seen that results of the 3-objective optimization process in the plane of (Pr_{resp}) and (S_∞^{-1}) successfully coincide with the boundary of the results of 4-objective optimization process which simply exhibit the Pareto front of these objective functions. It can be observed from the Pareto front of Figure (3) that improving one objective will cause another objective deteriorates accordingly.

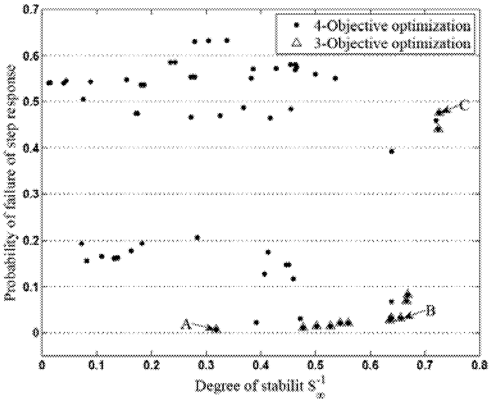


Figure 3: Pareto front of (Pr_{resp}) and (S_∞^{-1}).

The best point obtained for (Pr_{resp}) is point A which corresponds to the worst value of (S_∞^{-1}). These values are 0.0062 and 0.3184, respectively. Alternatively, the best value of obtained (S_∞^{-1}) is point C which corresponds to the worst value of (Pr_{resp}) and are 0.7259 and 0.4753, respectively.

The Pareto front of the 4-objective optimization process has also been shown on the plane of (Pr_{resp}) and ($\overline{u}$) of Figure (4).

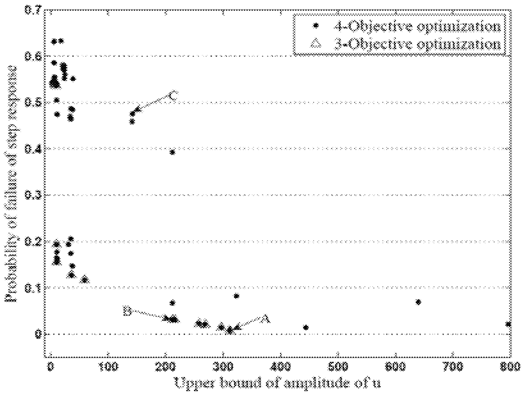


Figure 4: Pareto front of (Pr_{resp}) and ($\overline{u}$).

It can be observed from the Pareto front of Figure (4) that decreasing (Pr_{resp}) will cause ($\overline{u}$) increasing.

The numerical values of objective functions and the gain values of compensators for different optimum design points A, B, and C are shown in Table (1) and Table (2), respectively.

Table 1: Optimum values of objective functions.

Design points	Pr_{ins}	Pr_{resp} (%)	$\overline{u}$	S_∞^{-1}
A	0	0.62	311.54	0.3184
B	0	3.13	216.20	0.6555
C	0	47.53	142.70	0.7259

Table 2: Gain values for the compensators.

Design points	k_1	k_2	k_3	k_4
A	2.94	413.49	28.35	459.92
B	2.94	421.31	4.41	319.16
C	49.86	284.46	22.49	210.66

Design points	k_5	k_6	k_7	k_8
A	339.20	127.57	12.22	1.47
B	493.64	127.08	12.22	1.47
C	464.32	127.08	12.22	1.47

By careful investigation of Figure (3) an important trade-off can be observed from the Pareto front of objectives (Pr_{resp}) and (S_∞^{-1}). It is clear that the gradient of the Pareto front increases very gradually in section A-B. Apparently, optimum design point B shows a significant improvement in degree of stability (S_∞^{-1}) in comparison with that of point A whilst its probability of failure to the desired time response does not degrades significantly in section A-B as much as it does in section B-C. Figure (5) shows the corresponding 1, 10, 30, 50, 70, 90, 99 percentiles of time responses of design point B which demonstrates the stochastic behavior of the corresponding compensator for 500 Monte Carlo simulations of the plant subjected to the assumed probabilistic uncertainties. Thus, optimum design point B can be optimally chosen from a trade-off point of view for objectives (Pr_{resp}) and (S_∞^{-1}).

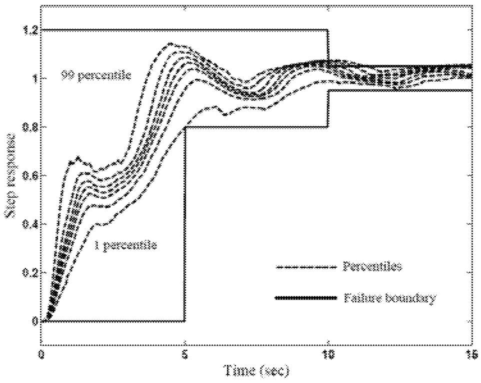


Figure 5: Step response behaviors of optimum design point B.

The robust stability margins of all optimum points (A, B, and C) have been shown in Figure (6). In this figure, the CDF have been shown for all design points. It is evident that the optimum design points B and C perform close to each other and both of them exhibit very good stability robustness.

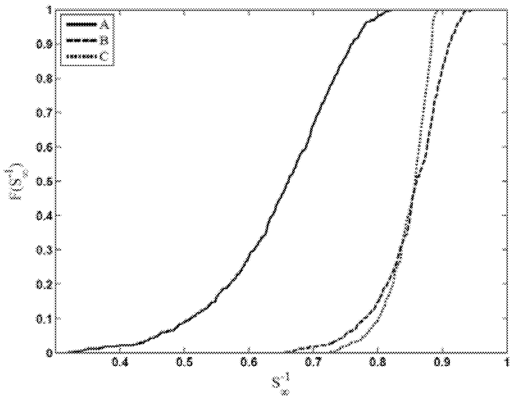


Figure 6: CDFs for robust stability margins of different optimum designs.

In order to evaluate the probability of instability more conservatively, the ranges of variation of uncertain parameters increased from ±50% to ±100% assuming same beta probabilistic distribution. The probability of instability has been computed using 10000 Monte Carlo simulations and the results of all points are shown in Figure (7).

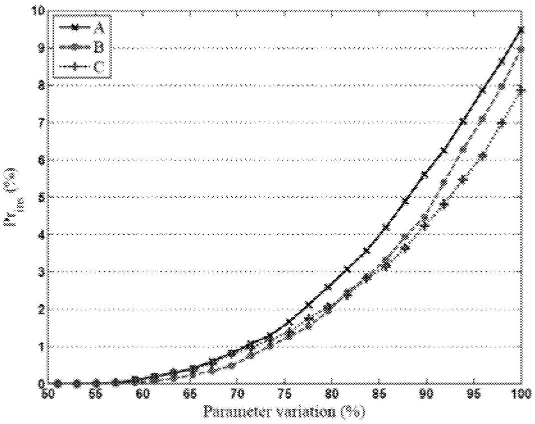


Figure 7: Probability of instability vs. the range of variations of uncertain parameters.

It is evident that all optimum points start experiencing instability around ±55% of parameter variations. The maximum value of (Pr_{ins}) for each design points A, B, and C are obtained at ±100% variation of parameters for the values of Pr_{ins} as 9.5%, 8.9%, and 7.8%, respectively.

CONCLUSION

A Pareto genetic algorithm was used to optimally design robust compensators for a dynamic system in a probabilistic approach. The objective functions which conflict with each other were appropriately defined using some probabilistic metrics both in time and frequency domains. The multi-objective optimization of robust compensators led to the discovering some important trade-offs among those objective functions. The framework of such application of multi-objective GAs of this paper for the Pareto optimization of both robust and reliability-based approach using some non-commensurable stochastic objective functions is very promising and can be generally used in the optimum design of real-world complex control systems with parametric uncertainties.

REFERENCES

Atashkari, K., Nariman-zadeh, N., Jamali, A., Pilechi, A., "Thermodynamic Pareto Optimization of turbojet using multi-objective genetic algorithm", International Journal of Thermal Science, Vol.44, No. 11, pp. 1061-1071, Elsevier, 2005.

Doyle, J.C., Francis, B.A., Tannenbaum, A.R., "Feedback Control Theory", Macmillan Publishing Company, 1992.

Goldberg, D.E., "Genetic Algorithms in Search, Optimization, and Machine Learning", Addison-Wesley, 1989.

Kang, Z., "Robust design of structures under uncertainties", PhD. Thesis, University of Stuttgart, 2005.

Lim, D., Ong, Y.s., Lee, B.S., "Inverse multi-objective robust evolutionary design optimization in the presence of uncertainty", GECCO' 05, Washington, USA, pp. 55-62, 2005.

Nariman-zadeh, N., Atashkari, K., Jamali, A., Pilechi, A., Yao, X., "Inverse modeling of multi-objective thermodynamically optimized turbojet engine using GMDH-type neural networks and evolutionary algorithms", Engineering Optimization, Vol. 37, pp. 437-462(26). 2005.

Papadrakakis, M., Lagaros, N.D., Plevris, V., "Structural optimization considering the probabilistic system response", Theoret. Appl. Mech., Vol. 31, No. 3-4, pp. 361-393, Belgrade, 2004.

Porter, B. and Jones, A.H., "Genetic tuning of digital PID controllers", Electronic Letters, 28(9), pp. 843-844, 1992.

Porter, B., Sangolola, A., Nariman-Zadeh, N., "Genetic design of computed torque controllers for robotic manipulators", IASTED Int. Conf. on Sys. and Control, Lugano, Switzerland, 1994.

Smith, B.A., Kenny S.P, Crespo, L.G., "Probabilistic Parameter Uncertainty Analysis of Single input Single Output Control Systems", NASA report, TM-2005-213280, March, 2005.

Stengel, R.F. "Stochastic Optimal Control: theory and application", New York, Wiley, 1986.

Stengel, R.F. and Ryan, L.E. "Stochastic robustness of linear control systems", Proc. Conf. Inform. Sci. Sys., pp. 556-561, 1989.

Wang, Q., Stengel, R.F., "Robust control of nonlinear systems with parametric uncertainty", Automatica, Vol. 38, pp. 1591 – 1599, 2002.

Wolovich, W.A., "Automatic Control Systems", Saunders College Publishing, Harcourt Brace College Pub., Orlando, USA, 1994.

GENERATING SIMULATION INPUT WITH APPROXIMATE COPULAS

Feras Nassaj
Johann Christoph Strelen
Rheinische Friedrich-Wilhelms-Universitaet Bonn
Institut fuer Informatik IV
Roemerstr. 164, 53117 Bonn, Germany

KEYWORDS

Simulation, Input Modeling, Dependency, Multivariate Random Numbers, Generation

ABSTRACT

Copulas are used in finance and insurance for modeling stochastic dependency. They comprehend the entire dependence structure, not only the linear correlations. Here they serve the purpose to analyze measured samples of random vectors, to estimate a multivariate distribution for them, and to generate random vectors with this distribution. This can be applied as well to time series.

INTRODUCTION

Stochastic models and discrete simulation are indispensable means for the quantitative analysis of systems. It is well known that missing to carefully model the influences from outside, especially the load, may lead to wrong results and ultimately to wrong decisions based on the simulation results. One reason for bad load models may be to ignore dependencies, i.e. to use independent random variables instead of proper commonly distributed random vectors or stochastic processes.

Influence from outside of the model like load or failure of system components can be incorporated into the model using observed traces or input models, namely random variables, random vectors, or stochastic processes. Data from traces can be used directly. If input is modeled, data are realisations of the model.

The use of random variates is well understood and common since long time, the use of generated random vectors and stochastic processes is much more difficult, not so popular, a topic of actual research.

In this paper, we propose to use copulas for the analysis of observed data and for the generation of dependent random variates and time series. The use of copulas is common in finance and insurance.

The copula of a multivariate distribution describes its dependence structure completely, not only the correlations of the random variables. It is uncoupled from the marginal distributions which can be modeled as empirical distributions or fitted standard distributions.

The use of copulas might make a difficult task, finding a multivariate distribution, more facile by performing two easier tasks. The first step is modeling the marginal distributions, the second consists in estimating the copula. Moreover, it is quite simple to generate random vectors with copulas.

In our approach, the marginal distributions might be modeled as empirical distributions or as theoretical distributions as usual. However, we estimate the copula as a frequency distribution, which is not common. Usually one of the known families of copulas is fitted. There are many such families, see e.g. (Nelsen 1998), but most of these families are for only two dimensions. For simulation, more dimensions might be needed. Moreover, as remarked in (Blum and Dias and Embrechts 2002), fitting a copula is essentially as difficult as estimating the joint distribution in the first place. Thirdly, different families of copulas account for different kinds of dependence. Hence, the input modeler must choose the family according to the actual dependence nature. In contrast, an empirical copula incorporates the dependence form automatically. For these reasons, we use some kind of empirical copulas (the frequency distribution) instead of fitting families of copulas.

A chi-square test is proposed for the evaluation of the goodness of the fitted approximate distribution.

The new technique contrasts with other proposed input models. For example, autoregressive processes (AR) model allow to model linear dependencies in time series with Gaussian random variables. They are conveniently fitted to measured data with the linear Yule-Walker equations.

ARTA-like models (ARTA (Cario and Nelson 1996) for univariate time-series, NORTA (Cario and Nelson 1997) for random vectors, VARTA (Biller and Nelson 2003) for processes of random vectors) allow also to model linear dependencies, more over, they allow for general distributions by means of a Gaussian AR or a multivariate Gaussian random variable as basis whose random variables are transformed into the desired distributions. The correlations of the basis process are different from the desired correlations. Therefore, a transformation is required. Sometimes this transformation results in unfeasible correlation matrices of the basis process (Ghosh

and Henderson 2002b), the *defective matrix problem.*
TES processes (Melamed 1997) rely on empirical distributions of the random variables. They comprise lag 1 correlations. The interactive software system TEStool serves the purpose of fitting measured data to a TES process.

AR, ARTA-like, and TES processes as input modeling approaches for random vectors and time-series consider only the linear correlations, not the whole dependence structure. In contrast, copulas take into account the entire dependency, hence this new technique as well.

In (Nassaj and Strelen 2005) we propose some kind of nonlinear non-Gaussian autoregressive processes. The dependence structure is more general, nonlinear dependencies are accounted for. The distributions of the random variables are general. The procedure of fitting to measured date is done in two successive steps. The first one for the dependence structure applies optimization with respect to some parameters. The second one concerns the distribution of univariate random variables. This separation is similar to the copula approach. However, this procedure requires knowledge or assumption about the type of dependencies. This problem does not appear in the approach we describe in this paper.

In the next section, some material about copulas is provided. Section 3 describes the procedure of building an approximate distribution which fits measured data, and how random vectors are generated from this distribution. The chi-square test for the evaluation of the distribution is given in section 4. Section 5 contains some examples.

COPULAS

A compact definition of copulas is given in (Pfeifer and Neslehova 2003):

Definition A *copula* is a function C of D variables on the unit D-cube $[0,1]^D$ with the following properties:
1. The range of C is the unit interval $[0,1]$
2. $C(\mathbf{u})$ is zero for all $\mathbf{u} \in [0,1]^D$ for which at least one coordinate equals zero
3. $C(\mathbf{u}) = u_d$ if all coordinates of $\mathbf{u}$ are 1 except the d-th one
4. C is D-increasing in the sense that for every $\mathbf{a} \leq \mathbf{b}$ in $[0,1]^D$ the measure ΔC_a^b assigned by C to the D-box $[\mathbf{a},\mathbf{b}] = [a_1,b_1] \times ... \times [a_D,b_D]$ is nonnegative, i.e.

$$\Delta C_a^b := \sum_{(\epsilon_1,\ldots,\epsilon_D)\in\{0,1\}} (-1)^{\epsilon_1+\ldots+\epsilon_D} C\Big(\epsilon_1 a_1 + (1-\epsilon_1)b_1, ..., \epsilon_D a_D + (1-\epsilon_D)b_D\Big) \geq 0.$$

In fact, a copula is a multivariate distribution function for the random vector $\mathbf{U} = (U_1, ..., U_D)$ with univariate uniform margins restricted to the unit D-cube. All partial derivatives exist almost everywhere, hence the conditional distribution functions and the density as well.

The key theorem due to Sklar clarifies the relations of dependence and the copula of a distribution:

Theorem (Sklar): Let F denote a D-dimensional distribution function with margins $F_1, ..., F_D$. Then there exists a D-copula C such that for all real $\mathbf{z} = (z_1, ..., z_D)$,

$$F(\mathbf{z}) = C\Big(F_1(z_1), ..., F_D(z_D)\Big).$$

If all the margins are continuous, then the copula is unique; in general, it is determined uniquely on the ranges of the marginal distribution functions. Moreover, if we denote by $F_1^{-1}, ..., F_D^{-1}$ the generalized inverses of the marginal distribution functions, then for every $\mathbf{u} = (u_1, ..., u_D)$ in the unit D-cube,

$$C(\mathbf{u}) = F\Big(F_1^{-1}(u_1), ..., F_D^{-1}(u_D)\Big).$$

For a proof, see (Nelsen 1998). In the next section, we define pseudo-inverses of distribution functions.

Multivariate random numbers $(z_1, ..., z_D)$ can be generated using copulas. First, we consider the special case of two dimensions:
1. Generate independent random numbers u_1 and u_2, uniform on $(0,1)$.
2. Use the pseudo-inverse of the conditional distribution function $C_2(u_2|U_1 = u_1) = P\{U_2 \leq u_2|U_1 = u_1\}$ for the generation of the random number u_2:

$$u_2 = C_2^{-1}(v|U_1 = u_1).$$

The conditional distribution function is equal to the partial derivative $\frac{\partial}{\partial u_1}C(u_1,u_2)$.
3. Univariate random variates can be generated with the inverse distribution function method $z_1 = F_1^{-1}(u_1)$ and $z_2 = F_2^{-1}(u_2)$. z_1 and z_2 are the elements of the desired random vector.

The generalization to higher dimensions is straightforward, the generation of the dependent u_d can be done in the usual way. See (Law and Kelton 2000),page 479, for example.

THE APPROXIMATE MULTIVARIATE DISTRIBUTION

We begin with a sketch of the method to fit an approximate multivariate distribution to data samples, the precise algorithm is presented subsequently.

A. Building the approximate distribution $\mathcal{A}$
1. Approximations $F_d(x)$, $d = 1, ..., D$, of the unknown marginal distribution functions are built from the given sample. This can be empirical distribution functions of some kind, or fitted standard distributions like exponential, Weibull etc.
2. The observed sample points $\mathbf{z}_i$, $i = 1, ..., n$, are transformed into points $\mathbf{u}_i$ of the unit D-cube $[0,1]^D$ by means of the marginal distribution functions.

3. The density of the approximate copula, that is, the density of the $\mathbf{u}_i$, is estimated. To this end, the D-cube is partitioned into sub-cubes. In each sub-cube, the density of the approximate copula is estimated from the number of points $\mathbf{u}_i$ in the subcube, divided by n and the volume of the sub-cube.

B. Generating random vectors.

In principle, using the approximate copula, random points $\hat{\mathbf{u}} \in [0,1]^D$ can be generated. From this, random vectors $\hat{\mathbf{z}}$ are obtained by means of the pseudo-inverses $F_d^{-1}(u) = \min\{z,\ F_d(z) = u\}$ of the estimated marginal distribution functions. Later on we indicate problems with this which occur if the marginal distribution functions are not continuous, and how to solve this problem.

Now we describe precisely the algorithm for the approximate distribution $\mathcal{A}$. Input data are a sample of random vectors $\mathbf{z}_i = (z_1, ..., z_D) \in \mathcal{Z}$, $i = 1, ..., n$, which are drawn from the unknown multivariate distribution $\mathcal{D}$ of the random vector $\mathbf{Z} = (Z_1, ..., Z_D)$.

Example 1. In Simulation, stochastic processes are more interesting. They can be analyzed and realized using a sliding window over a stochastic process. A special example is: $Z_{1,i} = A_i,\ Z_{2,i} = A_{i+1},\ i = 1, ..., n$, where $A_i,\ i = ..., -1, 0, 1, 2, ...$, is the stationary stochastic process defined by $A_{i+1} = 0.5\Big(1-4(A_i-0.5)^2\Big)+0.5X_i$, where the X_i are independent and uniformly distributed over $[0, 1]$.

Example 2, observed data at an Internet server. The data consists of inter-arrival times (A_i) and packet lengths (B_i). In this case, the stochastic process is $Z_{1,i} = A_i,\ Z_{2,i} = B_i,\ Z_{3,i} = A_{i+1},\ Z_{4,i} = B_{i+1},\ i = 1, ..., n$.

1. The empirical marginal distribution functions $F_d(z)$ are estimated. To this end, the sequences $z_{d,1}, z_{d,2}, ..., z_{d,n},\ d = 1, ..., D$, are ordered: $z_{d,(1)}, z_{d,(2)}, ...,$ where $i < j$ implies $z_{d,(i)} \leq z_{d,(j)}$. For $z_{d,(i)}$, where $z_{d,(i-1)} < z_{d,(i)} = z_{d,(i+1)} = ... = z_{d,(i+m-1)} < z_{d,(i+m)}$, $F_d(z_{d,(i)}) = m/n$ holds, $F_{d,i}$ for short (here we define $z_{d,(0)} = 0$ and $z_{d,(i,n+1)} = \infty$; these values are not used for estimation). For $z \in (z_{d,(i)}, z_{d,(i+1)})$, we define $F_d(z) = F_{d,i}$, but we will not use this. Alternatively, $F_d(z), d = 1, ..., D$, are fitted standard distributions.

2. Using the empirical or the standard marginal distributions, we get the transformed points $\mathbf{u}_i = (u_{1,i}, ..., u_{D,i}) \in [0,1]^D$, where $u_{d,i} = F_{d,i},\ d = 1, ..., D,\ i = 1, ..., n$.

3. For the sub-cubes, in each dimension d, the set $[0, 1]$ is partitioned into K_d subsets $S_{d,j}$ as follows: $S_{d,j} = [(j-1)\delta_d,\ j\delta_d),\ j = 1, ..., K_d-1,\ S_{d,K_d} = [(K_d-1)\delta_d,\ K_d\delta_d]$, where $\delta_d = 1/K_d,\ d = 1, ..., D$. With these subsets, the sub-cubes of $[0,1]^D$ are $\mathcal{S}_\mathbf{j} = S_{1,j_1} \times ... \times S_{D,j_D},\ \mathbf{j} \in \mathcal{K} = \{1, ..., K_1\} \times ... \times \{D, ..., K_D\}$.

Example. $D = 2$ dimensions, $K_1 = 3,\ K_2 = 4,\ \delta_1 = 1/3,\ \delta_2 = 1/4$

In a two dimensional cube, a partition $\mathcal{S}_\mathbf{j},\ \mathbf{j} \in \mathcal{K}$, induces a partition $\mathcal{T}_\mathbf{j} = T_{1,j_1} \times ... \times T_{D,j_D},\ \mathbf{j} \in \mathcal{K}$, in the original space $\mathcal{Z}$ of the observed random vectors $\mathbf{z}_i$ by means of $\mathbf{u} \in \mathcal{S}_\mathbf{j} \Leftrightarrow \mathbf{z} \in \mathcal{T}_\mathbf{j} \Leftrightarrow \forall d = 1, ..., D : z_d \in T_{d,j_d}$ where $\mathbf{u} = (u_1, ..., u_D)$, $\mathbf{z} = (z_1, ..., z_D)$, and $z_d = F_d^{-1}(u_d)$. See figure 1. This induced partition is unique only if the marginal distribution functions $F_d(z)$ are strictly increasing.

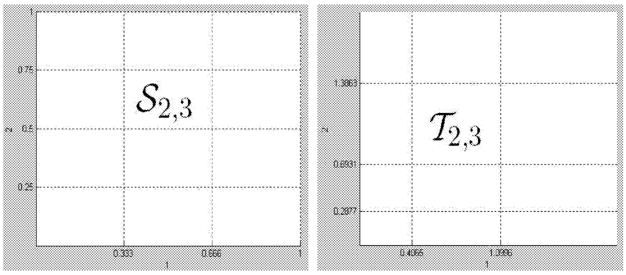


Figure 1: LEFT: SUB-CUBES OF THE UNIT D-CUBE. RIGHT: OF SPACE $\mathcal{Z}$, where $F_d(z) = 1 - \exp(-z)$

The approximate density of the copula is constant within each sub-cube $S_\mathbf{j},\ \mathbf{j} \in \mathcal{K}$. With the number $N_\mathbf{j}$ of points $u_\mathbf{j}$ in the sub-cubes $\mathcal{S}_\mathbf{j}$ and $H_\mathbf{j} = N_\mathbf{j}/n$, the density has the value $H_\mathbf{j}/(\delta_1 \cdot ... \cdot \delta_D)$. $H_\mathbf{j},\ \mathbf{j} \in \mathcal{K}$, is a frequency distribution for tuples $\mathbf{j}$. The reader may note that these approximations are not really a copula, in general: The marginal distributions are only approximately uniform. However, the empirical copulas, and thus their derivatives, the frequency copulas, converge to true copulas. See (Goorbergh and Genest and Werker 2005).

This (approximate) frequency copula, together with the pseudo-inverses of the marginal distributions, defines the approximate distribution $\mathcal{A}$. Its goodness of fit can be tested statistically with a chi-square test if a further sample of the same population is available. We present this in section 4.

Now we indicate how random vectors $\hat{\mathbf{z}}$ are generated form the fitted approximate distribution $\mathcal{A}$:

1. First a sub-cube $S_\mathbf{j}$ is selected randomly with equal probability $1/n$ according to the distribution $H_\mathbf{j},\ \mathbf{j} \in \mathcal{K}$, in the usual way, see for example (Law and Kelton 2000), page 479.

2. If the marginal distribution functions $F_d(z),\ d = 1, ..., D$, are continuous, a random point $\hat{\mathbf{u}} = (\hat{u}_1, ..., \hat{u}_D)$ is generated with a uniform distribution over the selected sub-cube. With the pseudo-inverses of the marginal distribution functions, the elements of the random vector are $\hat{z}_d = F_d^{-1}(\hat{u}_d),\ d = 1, ..., D$.

If the marginal distributions are discrete, we proceed differently. This is also the case for our empirical distribution functions. From all points $\mathbf{u}_i$ in the sub-cube, one point is selected randomly , say $\hat{\mathbf{u}}$. If one point occurs $m > 1$ times in the sample, the same point will be present several times in the sub-cube, say m-fold. In this case point probability is m/n.

For the transformation into the original space $\mathcal{Z}$ we use $F_d(z|Z_d \in T_{d,\hat{j}_d})$, the marginal distribution functions conditioned on Z_d lying in the interval $T_{d,\hat{j}_d}$ according to the selected sub-cube, for all dimensions d: $\hat{z}_d = F_d^{-1}(\hat{u}_d|Z_d \in T_{d,\hat{j}_d})$, $d = 1, ..., D$.

Why do we proceed differently for discrete distributions? Consider the following situation. Let $u' = F_d(z_{d,(i-1)}) < j\delta_d < F_d(z_{d,(i)}) = u''$, $\hat{u} = u'(1+\epsilon)$, $\epsilon > 0$ so small that $\hat{u} < j\delta_d$. Hence, $u' \in S_{d,j}$, $u'' \in S_{d,j+1}$ and $z_{d,(i-1)} \in T_{d,j}$, $z_{d,(i)} \in T_{d,j+1}$. If in the generating process a sub-cube $... \times S_{d,j} \times ... \in [0,1]^D$ was selected and $\hat{u}$ was generated for $\hat{z}_d$, the $\hat{\mathbf{z}}$ which is generated via $\hat{z}_d = F_d^{-1}(\hat{u}) = z_{d,(i)}$ would lie in the sub-cube $\mathcal{T} \in \mathcal{Z}$ which corresponds to the sub-cube $... \times S_{d,j+1} \times ... \in [0,1]^D$, not $... \times S_{d,j} \times ... \in [0,1]^D$ which was determined by the algorithm. This problem could alternatively be omitted with a different definition of the sub-cubes.

The computational cost for the method is $O(n \log n + nK_1 \cdot ... \cdot K_D)$. In our examples, calculated with MATLAB on a 1GHz PC, the computing times were seconds or few minutes.

A CHI-SQUARE TEST FOR THE QUALITY OF THE APPROXIMATE DISTRIBUTION

In this section, we compare the approximate distribution $\mathcal{A}$ with a second sample $\mathbf{z}'_i$, $i = 1, ..., n'$, from the same population as the sample $\mathbf{z}_i$, $i = 1, ..., n$, which we used to build $\mathcal{A}$, by means of a chi-square goodness-of-fit test. If the hypothesis is not rejected, we take this as an indication of the quality of $\mathcal{A}$.

For the chi-square test the sample must consist in independent points, but this is not fulfilled in general. Therefore we start with a larger sample and discard points between $\mathbf{z}'_i$ and $\mathbf{z}'_{i+1}$ and hope that spaced points are nearly independent.

For the test, the space $\mathcal{Z}$ must be partitioned. We use the partition $\mathcal{T}_\mathbf{j}$, $\mathbf{j} \in \mathcal{K}$, but in each subset must be enough points of the first sample. This is in general not the case. Therefore we combine sub-cubes to a subset $\mathcal{R}_l$ until $n' \cdot p_l \geq 5$ where p_l is the sum of the probabilities $H_\mathbf{j}$ of all combined sub-cubes $\mathcal{T}_\mathbf{j}$; this is a usual heuristic. Thus we obtain r subsets $\mathcal{R}_l$ and probabilities p_l, $l = 1, ..., r$. The precise definition of this combination is provided by an algorithm in the appendix.

As we said before, the partition $\mathcal{T}_\mathbf{j}$, $\mathbf{j} \in \mathcal{K}$, is not unique, in general. Here we define the intervals in each dimension as follows: $T_{d,j} = [l_{d,j}, h_{d,j})$ for all d and all $j = 1, ..., K_d$, where $l_{d,1} = 0$, $l_{d,j} = F_d^{-1}(u)$ with $u = \min_{1 \leq i \leq n}\{u_{d,i} \in \mathcal{S}_{d,j}\}$, $h_{d,j} = l_{d,j+1}, j = 1, ..., K_d - 1$, and $h_{d,K_d} = \infty$.

Let $N'_\mathbf{j}$ denote the number of points $\mathbf{z}'_\mathbf{j}$ in the sub-cube $\mathcal{T}_\mathbf{j}$ and y_l the number in the subset $\mathcal{R}_l$, $l = 1, ..., r$.

The test statistic is the χ^2-distance function $Q = \sum_{l=1}^r y_l^2/(n'p_l) - n'$ which is compared with the $(1-\alpha)$-quantil of the χ^2-distribution with $r-1$ degrees of freedom.

EXAMPLES

In the numerical examples, correctness and accuracy are verified with the chi-square test, with some statistics and, visually, with scatter diagrams.

Statistics and diagrams are calculated for the measured sample and time series which are generated with the approximate distribution $\mathcal{A}$. The statistics are means, coefficients of variation, and correlations of the $z_{d,i}$, the latter between $z_{d,i}$ and $z_{d',i}$, $d \neq d'$. First we calculated the differences of corresponding coefficients of variation and correlations, and relative differences of corresponding means. In order to not bother the reader with many figures, we only give the maximum of the absolute values of these differences, the *maximum statistics difference*.

The method primarily serves the purpose to analyze the multivariate distribution of random vectors and to generate random vectors for simulation input. In our examples, it is indicated how it can be used for time series.

Under these circumstances, when the next $\hat{\mathbf{z}}_i$ is generated, some elements from the previous $\hat{\mathbf{z}}_{i-1}$ can be taken, e.g. in example 2, $\hat{z}_{1,i} = \hat{z}_{2,i-1}$. If for the generation of $\hat{\mathbf{z}}_{i-1}$ the sub-cube was $\mathcal{T}_\mathbf{j}$, then for the next generation, the interval T_{1,k_1} of the next sub-cube $\mathcal{T}_\mathbf{k}$ equals T_{2,j_2} from the previous sub-cube.

Here, two kinds of errors may occur. First, if the following holds: In the sample $\mathbf{z}_i$, $i = 1, ...n$, from which the distribution $\mathcal{A}$ was built, there is no point $\mathbf{z}_i$ which lies in any sub-cube $\mathcal{T}_\mathbf{k}$ with $T_{1,k_1} = T_{2,j_2}$. That means the row $H(j_2, .)$ has only zero entries. Hence, in the selected sub-cube, no point can be generated, the generation leads into a *dead end*.

Secondly, the probabilities $H_\mathbf{j}$ can be so that the generated points end in a cycle of points which recur again and again.

These errors occur with some probability if the sample is small, in bigger samples this probability becomes smaller and smaller. In fact, we observed these problems only when very small samples were used for the distribution $\mathcal{A}$, ten points or so.

If a bigger sample is impossible or unwanted, there is a remedy: When a dead end or a cycle occurs. $\hat{\mathbf{z}}_i$ is generated completely new under violation of $\hat{z}_{1,i} = \hat{z}_{2,i-1}$.

Example 1

We consider a sliding window over a stochastic process. $Z_{1,i} = A_i$, $Z_{2,i} = A_{i+1}$, $i = 1, ..., n$, where A_i, $i = ..., -1, 0, 1, 2, ...$, is the stationary stochastic process defined by $A_{i+1} = 0.5\Big(1 - 4(A_i - 0.5)^2\Big) + 0.5X_i$, where the X_i are independent and uniformly distributed over $[0,1]$.

Sample size $n = 4000$. The number of subintervals in each of the two dimensions $K_1 = K_2 = 40$. The maximum statistics difference is 0.018, no dead end occurred. The scatter diagrams of the sample and the generated points indicate that there are probably regions where no points can exist, and that these regions are observed by the generated process with good accuracy, as seen in figure 2. For comparison, we generated a process with a fitted linear nGAR model. Obviously, here many generated points lie in impossible regions as seen in figure 3. For four different streams of random numbers, we built the approximate distribution $\mathcal{A}$ with small samples, $n = 100$, and larger samples, $n = 400$, and $K_1 = K_2 = 20$. For the small sample sizes, the chi-square test indicated three of four times "reject", for the larger sample sizes not once.

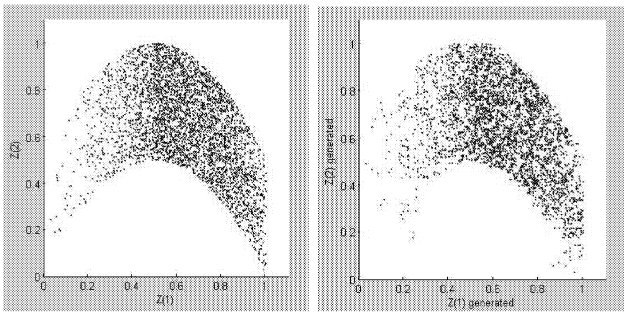


Figure 2: LEFT: THE ORIGINAL SAMPLE. RIGHT: THE GENERATED PROCESS

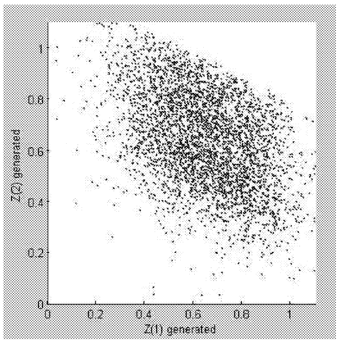


Figure 3: PROCESS GENERATED WITH AN AR MODEL

Example 2

We consider observed data by (Klemm, Lindemann and Lohmann 2002) at an Internet server. The data consists of inter-arrival times (A_i) and packet lengths (B_i).
In this case, the stochastic process is $Z_{1,i} = A_i,\ Z_{2,i} = B_i,\ Z_{3,i} = A_{i+1},\ Z_{4,i} = B_{i+1},\ i = 1, ..., n.$
Sample size $n = 4000$, $K_1 = K_2 = K_3 = K_4 = 40$. The maximum statistics difference is 0.03, no dead end occurred. The scatter diagrams of the sample and the generated points indicate good accuracy. See figures 4 and 5. Figure 6 is a scatter diagram of the points $(u_{1,i}, u_{2,i})$, hence some visualization of the marginal density of the copula, dimensions 1 and 2. Figure 7 visualize the dimension 1 and 2 of the sub-cubes $\mathcal{S}_\mathrm{j}$ for $K_1 = K_2 = K_3 = K_4 = 40$ and $K_1 = K_2 = K_3 = K_4 = 60$. Every point indicates one or more sub-cubes with sample points. Obviously, the higher accuracy separates better the impossible regions.

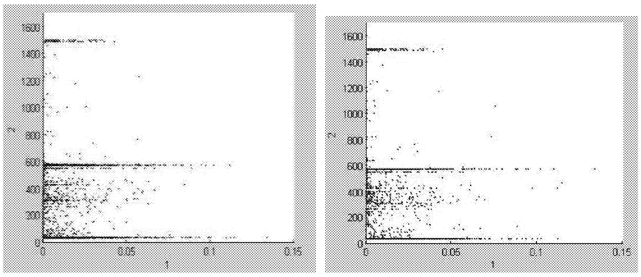


Figure 4: LEFT: THE ORIGINAL SAMPLE. RIGHT: THE GENERATED PROCESS; DIMENSIONS 1 & 2

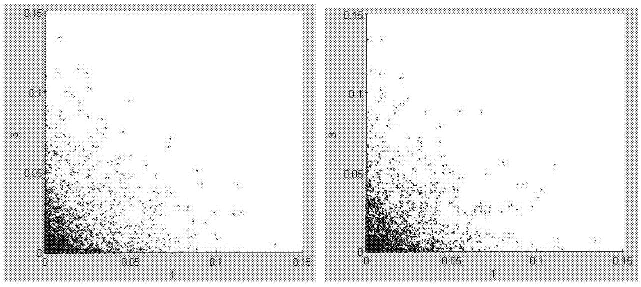


Figure 5: LEFT: THE ORIGINAL SAMPLE. RIGHT: THE GENERATED PROCESS; DIMENSIONS 1 & 3

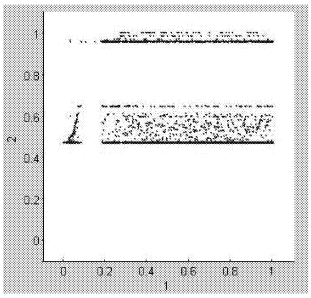


Figure 6: TRANSFORMED SAMPLE POINTS, $(u_{1,i}, u_{2,i})$

CONCLUSION

Copulas seem to be useful for the analysis of multivariate samples and for the generation of multivariate random numbers and time series. In contrast to other approaches, this approach is able to approximate any type

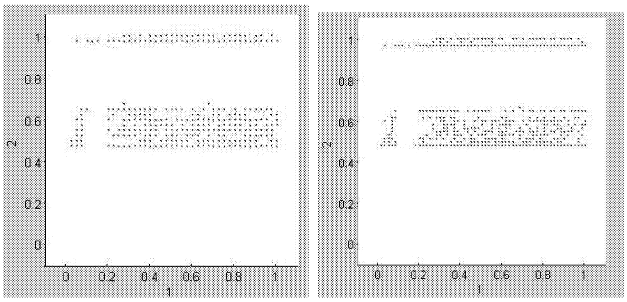


Figure 7: DIMENSIONS 1 AND 2 OF THE SUB-CUBES, DIFFERENT APPROXIMATION ACCURACY

of models (linear or nonlinear regression, multidimensional time-series, ...).
For future work, more goodness-of-fit tests can be done, and variations of the proposed method should be considered, for example:

- More flexible sub-cubes.
- Other marginal distribution functions, e.g. fitted standard distributions.
- Other kinds of estimated copulas.

Acknowledgement We gratefully appreciate the recommendation of our colleague Dr. H.J. Kühn to consider copulas.

REFERENCES

Biller B. and B. L. Nelson, 2003, “Modeling and generating multivariate time-series input processes using a vector autoregressive technique,” *ACM Transactions on Modeling and Computer Simulation*, vol. 13, no. 3, pp. 211–237.

P. Blum and A. Dias and P. Embrechts, 2002, “The art of dependence modelling: the latest advances in correlation analysis,” *Alternative Risk Strategies*, London. 339-356.

Cario M. C. and B. L. Nelson, 1996, “Autoregressive to anything—time-series input processes for simulation,” *Operations Research Letters*, vol. 19, no. 2, pp. 51–58.

Cario M. C. and B. L. Nelson, 1997, “Modeling and generating random vectors with arbitrary marginal distributions and correlation matrix”, *Department of Industrial Engineering and Management Sciences*, Evanston, Ill.

F. Nassaj and J. Ch. Strelen, 2005, “Dependence input modeling with the help of non-Gaussian AR models and genetic algorithms,” *Modelling and Simulation 2005, Proceedings of the European Simulation and Modelling Conference, Porto, 2005*, pp. 146-153.

Ghosh S. and S. G. Henderson, 2002b, “Properties of the norta method in higher dimensions,” in *Winter Simulation Conference Proceedings*, Piscataway, N.J., pp. 263–269.

Klemm A., C. Lindemann, and M. Lohmann, 2002, “Traffic Modeling of IP Networks Using the Batch Markovian Arrival Process”, *12th Int. Conf. on Modelling Tools and Techniques for Computer and Communication System Performance Evaluation.* London. UK. Lecture Notes in Computer Science, 2324, 92-110.

Law A. M. and D. W. Kelton, 2000, *Simulation Modeling and Analysis, 3rd edition.* New York: McGraw-Hill.

B. Melamed, 1997, “The empirical TES methodology: Modeling empirical time series,” *J. of Applied Mathematics and Stochastic Analysis*, vol. 10, no. 4, pp. 333-353.

R.B. Nelsen, 1998, *An introduction to copulas*, New York: Springer.

D. Pfeifer and J. Neslehova, 2003, “Modeling dependence in finance and insurance: the copula approach,” *Blätter der DGVFM*, vol. 26, no. 2, pp. 177-191.

Rob W.J. van den Goorbergh, Christian Genest, Bas J.M. Werker, 2005, “Bivariate option pricing using dynamic copula models,” *Insurance: Mathematics and Economics*, vol. 37, no. 1, pp. 101-114.

APPENDIX

Algorithm for the subsets of the chi-square test

```
$l := 0$;
$sum := 0$;
$sum' := 0$;
for all $\mathbf{j} \in \mathcal{K}$
    $sum$ plus $n'H_{\mathbf{j}}$;
    $sum'$ plus $N'_{\mathbf{j}}$;
    if $sum \geq 5$
        $l$ plus 1;    // next subset $\mathcal{R}_l$
        $n'p_l := sum$;
        $y_l := sum'$;
        $sum := 0$;
        $sum' := 0$;
    fi
end
$n'p_1$ plus $sum$;
$y_1$ plus $sum'$;
$r := l$;    //number of subsets
```

EXPANDED SCOPE OF TRAFFIC-FLOW ANALYSIS: ENTITY FLOW-PHASE ANALYSIS FOR RAPID PERFORMANCE EVALUATION OF ENTERPRISE PROCESS SYSTEMS

Gábor Lencse
Department of Telecommunications
Széchenyi István University
Egyetem tér 1.
H-9026 Győr, Hungary
e-mail: lencse@sze.hu

László Muka
Elassys Consulting Ltd.
Bég utca 3-5.
H-1026 Budapest, Hungary
e-mail: muka.laszlo@elassys.hu

KEYWORDS

entity flow-phase analysis, traffic-flow analysis, information and communication technology, organisational process, process modelling, parallel simulation, discrete event simulation

ABSTRACT

This paper describes entity-flow phase analysis (EFA) which is a method for fast performance analysis of organisational process systems. EFA, similarly to traffic-flow analysis for communication systems (TFA), uses the combined approach of simulation and numerical methods. In simulation projects initiated to support the design of Information and Communication Technology (ICT) system and Business Process (BP) system in an organisation the parallel analysis of different systems may be efficient. EFA is a promising evaluation method to be applied for systems with determined BP and ICT subsystems in an organisational environment.

INTRODUCTION

Mixed simulation projects

Simulation projects aimed to support the design of Information and Communication Technology (ICT) and Business Process (BP) systems in an organisation traditionally are independent, separate projects, in spite of the fact that these systems may have significant influence on each other. Common analysis of these systems may have advantages but in this case we need to have methods appropriate for both types of systems.

Process system definition

There are some known, basic definitions of business processes:

By the definition given in (Davenport 1993) processes are structured, measured sets of activities designed to produce a specified output for a particular customer or market.

According to another definition a business process is a partially ordered set of Enterprise Activities which can be executed to realise a given objective of an enterprise or a part of an enterprise to achieve some desired end-result (Savén 2002).

In enterprise (organisational) modelling business process is defined as a network of actions performed in context of one or more organisational roles in pursuit of some goal (Koubarakis and Plexousakis 1999)

According to the above requirements we give a definition to be used to our modelling purposes:

Business processes are related to enterprises and they define the way in which the goals of the enterprise are achieved.

Business Process is a set of Enterprise Activities linked together to form a process with one or more kinds of input to produce outputs.
A process system is a set of business processes linked together to perform some Enterprise Function or Subfunction.
(Processes of an enterprise can be identified using process mapping (Graesley 2000)).

The Traffic-Flow Analysis

The *traffic-flow analysis* (TFA) (Lencse 2001, about the convergence of TFA: (Lencse and Muka 2006)) is a simulation-like method for the fast performance analysis of communication systems. TFA uses statistics to model the networking load of applications.

In the first part the method distributes the traffic (the statistics) in the network, using routing rules and routing units.

In the second part the influence of the finite capacities (line and switching-node capacities) is calculated.

The important features of TFA:

The results are approximate but the absence and the place of bottlenecks is shown by the method.

The execution time of TFA is expected to be significantly less than execution time of detailed simulation of the system.

TFA describes the steady state behaviour of the network (there is no need for warm-up time definition).

IDENTIFYING EFA MODEL ELEMENTS

EFA: A new evaluation method

In this paper we introduce a new method, EFA for the fast process analysis. EFA is based on the experience of TFA.

We introduce two versions of EFA:

1. In the first version the analysis is performed in two steps: first the spatial distribution of entity-load is determined, then the time distribution is calculated (One-phase Method)
2. In the second version the analysis is performed by repeating the steps of determination of spatial distribution of entity-load and the time distribution is calculation for activity groups featured with equal distance from entity-load source (Multi-phase Method)

We may have a promising capability of common analysis of BP and ICT systems using EFA-TFA methods.

An overview of TFA model elements

Now, before identifying EFA model elements we summarise TFA model elements:

Network model of TFA consists of nodes (routers, switches, etc.) and lines (transmission lines).

Traffic model: TFA uses probability density functions (PDF) to model the traffic load: PDF of throughput and PDF of delay. (The traffic is generated by applications (application models)). The delay distributions are calculated when the influence of finite line and node capacities are taken into account.

(Any traffic model can be used in TFA (mathematical or statistical) that satisfies the requirements described in (Lencse 2001)).

Throughput is the number of packets or bits arrived in a time interval ***T***. It is clear that the value of ***T*** has significant influence on the distribution.

Bit-throughput and packet-throughput distributions are used to describe the traffic on the lines and in the nodes.

The routing model (which can be any) of the network is used to distribute the traffic.

EFA model elements: Activity model-element, linking activities

By definition the process is a set of activities which are linked together.

The links are connections with a direction showing the performance order (time-precedence) of activities. Internal links are connecting the activities of one process; external links are connecting processes forming a process system. The links are only logical connections with no capacity limit.

Performing an activity takes time which can be better described by a probability distribution (service time profile, activity time-consumption)**,** than by an exact value because many factors are influencing the performance-time (for example the learning process of the assigned process resource). In many cases it is enough to use normal distribution and the expected value of activity time:

T_{Cons}(activity, entity type) Time-Consumption of an Activity – expected value of time necessary to perform the given activity, that is necessary to process the entering entity-type.

EFA model elements: Entity-load model

The entity-load in the model is produced by programmable entity-generator, the source of incoming entities. There may be different types of entities entering the process, which are produced by different sources, generators.

Entity-load models to be used:

1. An entity ***arrival profile*** describes the arrival time distribution of entity-load for an entity of a given type.

2. TFA-like ***entity-throughput*** model uses the probability distribution describing the entity-load intensity: the probability that ***k*** entities directed to an activity to be processed in a time interval of ***T*** length is ***$p_T[k]$.*** In the steady state ***$p_T[k]^*$*** denotes that exactly ***k*** entities need to be serviced in ***T*** time. (Consideration about the value of ***T,*** see below.)

The entity-load model is taken as an input for the model, while the delay-time is calculated.

The delay caused by the resource capacity limit can be calculated for the entity-throughput model using a formula similar to TFA delay-time calculation:

$$D_T[i] = \begin{cases} \sum\limits_{k=0}^{R_n} p_T[k]^*, i = 0 \\ \sum\limits_{k=iR_n+1}^{(i+1)R_n} p_T[k]^*, i \geq 1 \end{cases}$$

where ***$D_T[i]$*** is the probability of an ***i*T*** delay, ***R_n*** is the resource capacity limit for activity ***n***.

(Remark: a more precise calculation can be done using the Bayes' theorem and the probability distribution of resource accessibility.)

The destinations of entities are exit (result) points of the process. The output observations of the process can be made and the necessary output statistics can be created at these points.

EFA model elements: The T interval

The ***T*** interval is the resolution of our examination.

(It is also clear that the entity-load intensity distribution influenced by the determination of ***T***.)

In process analysis a typical value of ***T*** is an hour, but in the assessment of Callcenters the typical value is one minute. In examination of ICT-BP connections it may be necessary to use seconds.

EFA model elements: Entity routing, process decisions

Similarly to communication systems there is a routing of entities in the process. Entity routing depends on process decisions.

A routing decision may be made using different algorithms:

Percentage distribution: the destination of an entity is decided by the probability distribution of the possible entity outputs.

Entity-feature distribution: output for an entity is chosen by some feature of an entity (type, priority, etc.).

Load-balancing distribution: output for an entity is chosen on the basis of some load-balancing algorithm (including some quantity-limit consideration too).

Other details

There are some other elements influencing the generation and routing of entities:

Fork element makes copies of an entity (this is a parent-child relationship) and routes the copies to outputs in a parallel way. The Fork's pair is the Join element that collects the entities divided by Fork element into one entity. The delay of an entity collected by Join equals the maximum delay of entities routed by Fork.

Split element also makes copies of an entity and routes the copies according to the output links of the element but the splitted entities will not be collected into one entity again. Split generates entities, which will have separate ways in the process.

Transform element may change the entity's features in the process**.**

A CAPACITY-LIMIT MODEL

In the following, we determine a Resource Capacity-Limit Model (RCLM) to be used by EFA method.

RCLM has two important basic features:

1. RCLM should describe different groups of resources, where a group functions as a pool of resources, having a summary limit for the resources in the pool.
2. If a given resource element is engaged in one activity of the process it cannot be used by another one at the same time.

The RCLM elements and parameters:

List of resource types describing the resources required in the process

- $\boldsymbol{N_{Res}}$ ***(type)*** Number of Resource-Elements of a given type – the number is the capacity limit of the resource group (pool)

List of activities using a given type of resource

- $\boldsymbol{P_{ARes}}$***(type, month, week, day, time)*** Resource Accessibility – the probability that a resource of the given type is accessible at the given point of time for a given activity, in an interval of ***T length***. (For the correct description of the "resource behaviour" we should examine longer periods, to take into consideration the seasonality.)
- $\boldsymbol{R_n}$ - expected value of accessible resource capacity of a given type for an activity ***n*** (non-negative integer). We may use $\boldsymbol{R_n}$ to decrease the amount of calculations.

THE EFA METHOD

The work of the EFA method, similarly to TFA, can be divided into two parts:

1. Distribution of the entity-load in the process
Sending entity-load statistics to the process activities according to the routing conditions
At this point summation of statistics must be performed.

2. Calculation of time influences of finite capacities of resources

We may use the expected value of a resource for an activity or the probability distribution of resource accessibility can be used to get more precise result.

Remark: if there are feed-back loops in the process first they should be eliminated: it may be done by adding a calculated portion of entity-load of the output point of feed-back to its input point and then cutting-off the loop.

M1 One-phase method

M1.1 Sending statistics to every process activity according to the routing
M1.2 Calculate the processing-time of activities parallel.
(The execution of the second step is expected to require much more processing power then the exection of the first step, that's why the second step may worth executing parallel by multiple processors.)

M2 Multi-phase method

M2.1 Sending statistics to all process activities which are in equal distance from process entity-load source (starting with the nearest group of activities). Sending statistics is performed according to the routing rules.
M2.2 Calculate the processing-time for activities in the group in a parallel way.
M2.3 Repeat cycle M2.1 and M2.2 for the next nearest group of activities.
Remark: if there are feed-back loops in the process, first a One-phase method run should be applied to eliminate feed-back.

TESTING THE EFA APPROACH: AN APPLICATION EXAMPLE

Let us see an example to compare process analysis methods: the application of Event-driven Discrete-Event Simulation (DES) and EFA.

The test-process topology (elements and links) is shown in Figure 1. The process has two sources for generation of entity-load: Source 01. and Source 02. There are 7 activities in the process: Activity 01-07. There is one exit-point in the process: Result 01.

The entity-load model is described by two arrival profiles with normal distributions. The Activities 01-07 have different service profiles which are also described by normal distributions.

The resources for all activities are placed into one pool of resources with high capacity limit: the expected value of accessible resource capacity is higher than the number of incoming entities require.

The routing decision may be made using percentage distribution at outputs of Activity 02.

The resolution (T) is 0.01 day. The interval of observation (simulated time) is 365 days.

There are two process models built: a Flow-DES simulation model (F-DES) and a Multi-phase EFA model (EFA-M).

(Both process models (F-DES and EFA-M) are created in an Event-driven Discrete-Event Flow-Simulator, ImiFlow™ of Elassys Consulting Ltd.)

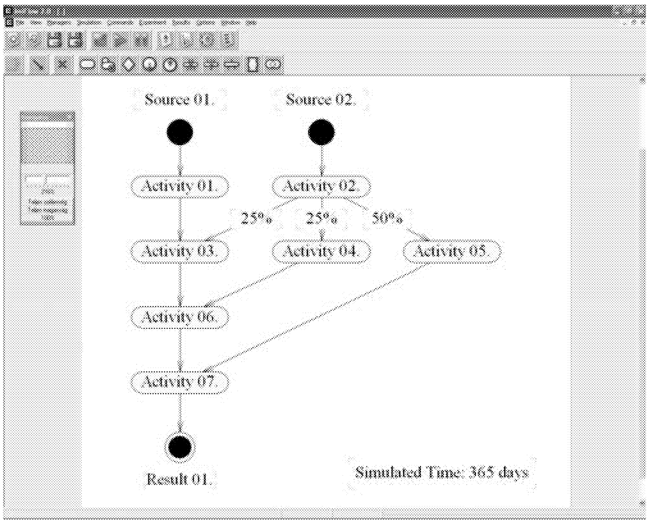


Figure 1. The examined test enterprise process

The comparison results are summarised in Figure 2., Figure 3. and Table 1.

Figure 2. shows the delay time (measured in days) of entities at Activity 05. for DES and EFA methods. There is a higher dynamics in DES but the moving averages (thick dashed lines) are closer to each other.

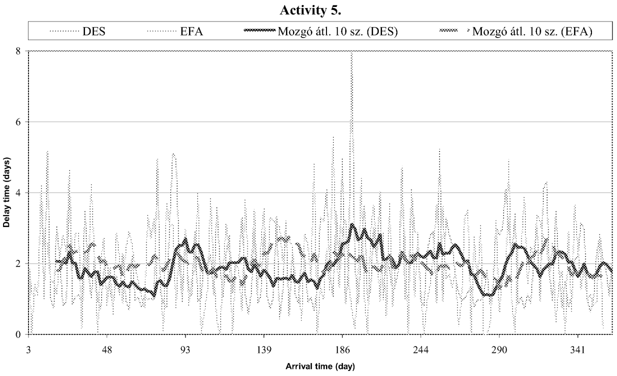


Figure 2. The output delay at Activity 05.

Figure 3. shows the delay time of entities at Activity 07. which is the exit point of the process. Here, the arrival frequency of entities is higher than in Figure 2. DES and EFA moving averages are again close to each other.

The average delays for DES and EFA for every Activity and for the whole observation period are collected in Table 1. The results in columns F-DES and EFA-M are highly correlated.

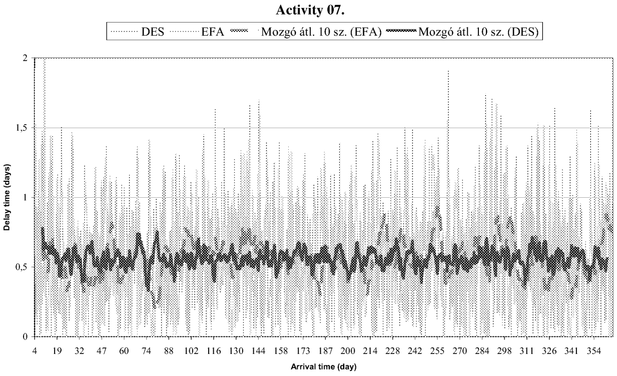


Figure 3. The output delay at Activity 07.

Name	F-DES	EFA-M
Activity 01.	1.52 day(s)	1.50 day(s)
Activity 02.	1.30 day(s)	1.30 day(s)
Activity 03.	3.21 day(s)	3.19 day(s)
Activity 04.	3.18 day(s)	3.46 day(s)
Activity 05.	2.21 day(s)	2.16 day(s)
Activity 06.	4.46 day(s)	4.73 day(s)
Activity 07.	5.89 day(s)	5.85 day(s)

Table 1. Comparison of DES and EFA delays

CONSIDERATIONS ABOUT THE APPLICATION OF EFA FOR PARALLEL ANALYSIS

In the case of information system design in an organisation, after identifying the ICT and BP subsystems to be examined (using the meta-methodology developed by the authors) we may have different situations:

If we have one ICT and one BP subsystem, depending on the focus of the simulation there can be three basic parallel simulation decisions: (1) detailed simulation of both ICT and BP subsystems; (2) detailed simulation of ICT system with simulated BP as process environment; (3) detailed simulation of BP with simulated ICT system as environment. The BP and ICT parts can act as the two segments of parallel discrete event simulation. They can be executed parallel by two interconnected processors. For all the three situations the use of the Statistical Synchronisation Method (Pongor 1992) can be considered as an inter-processor synchronisation method if there is a relatively slow speed of changes in subsystems' states. ***In situation (2) the use of EFA may be considered.*** In situation (3) the method of TFA may be appropriate.

Methods for the ***parallel execution of the Combined DES and TFA*** (Lencse 2004) can be found in (Lencse 2005); similar considerations can also be made for ***EFA*** application. (Combined application of DES, TFA and EFA should be the object of future research work.)

CONCLUSIONS

We have introduced a new method for the fast performance analysis, EFA in this paper.

We have defined model elements for EFA:

- we have described activity model-element and linking of activities,
- we have defined entity-load models: the usual arrival profile, and the entity-throughput model,
- we have given a formula and a method for delay-time calculation,
- we have examined the problem of entity routing depending on process decisions.

Using the introduced elemets we have outlined two versions of EFA method: One-phase method for rapid analysis and Multi-phase method for a more precise fast evaluation.

We have given solution to the problem of possible feed-back loops in the examined process.

In the end we have tested the EFA method on an example of an enterprise process.

ACKNOWLEDGEMENTS

The authors would like to thank Gergely Muka for the valuable help in the process simulation and Peter Lukács for the support in statistics.

REFERENCES

Greasley, A. 2000. *Effective Uses of Business Process Simulation* Proceedings of the 2000 Winter Simulation Conference, Joines, J. A., Barton, R. R., Kang, K., Fishwick, P. A., eds.

Koubarakis, M., Plexousakis, D. 1999. *Business process modelling and design – a formal model and methodology* BT Technol. J. Vol. 17, No. 4.

Davenport, T.H. 1993. *Process innovation: Reengeneering work through information technology* Harvard Business School Press, Boston, Massachusetts

Savén, R. 2002. *Process Modelling for Enterprise Integration: review and framework* Department of Production Economics, Linköping Institute of Technology, Linköping, Sweden

Elassys Consulting Ltd. 2004. "ImiFlow System" http://www.elassys.hu

Lencse, G. 2001. *Traffic-Flow Analysis for Fast Performance Estimation of Communication Systems* Journal of Computing and Information Technology 9, No. 1, pp. 15-27.

Lencse, G. 2004. *Combination and Interworking of Traffic-Flow Analysis and Event-Driven Discrete Event Simulation* Proceedings of the 2004 European Simulation and Modelling Conference (ESM®'2004) (Paris, France, Oct. 25-27. 2004.) EUROSIS-ETI, 89-93.

Lencse, G. 2005. *Speeding up the Performance Analysis of Communication Systems* Proceedings of the 2005 European Simulation and Modelling Conference (ESM®'2005) (Porto, Portugal, Oct. 24-26.) EUROSIS-ETI, 329-333.

Lencse, G., Muka, L. 2006. *Convergence of the Key Algorithm of Traffic-Flow Analysis* Journal of Computing and Information Technology Vol 14, No. 2, pp.133-139

Pongor, Gy. 1992. *Statistical Synchronization: a Different Approach of Parallel Discrete Event Simulation* Proceedings of the 1992 European Simulation Symposium (ESS'92) (Nov. 5-8, 1992, The Blockhaus, Dresden, Germany.) SCS Europe, 125-129.

BIOGRAPHY

GÁBOR LENCSE received his M.Sc. in electrical engineering and computer systems at the Technical University of Budapest in 1994 and his Ph.D. in 2000. The area of his research is (parallel) discrete-event simulation methodology. He is interested in the acceleration of the simulation of communication systems. Since 1997, he works for the Széchenyi István University in Győr. He teaches computer networks and networking protocols. Now, he is an Associate Professor. He does R&D in the field of the simulation of communication systems for the Elassys Consulting Ltd. since 1998. Dr Lencse works part time at the Budapest University of Technology and Economics (the former Technical University of Budapest). There he teaches digital design and computer architectures.

LÁSZLÓ MUKA graduated in electrical engineering at the Technical University of Lvov in 1976. He got his special engineering degree in digital electronics at the Technical University of Budapest in 1981, and became a university level doctor in architectures of CAD systems in 1987. Dr Muka finished an MBA at Brunel University of London in 1996. Since 1996 he has been working in the area of simulation modelling of telecommunication systems, including human subsystems. He is a regular invited lecturer in the topics of application of computer simulation for performance analysis of telecommunication systems, at the Széchenyi István University of Győr.

ANALYTICAL AND NUMERICAL SIMULATION IN COMMUNICATIONS

DIFFERENTIAL MODELING AND ITS APPLICATION TO TCP/IP

H. Hassan, J-M.Garcia and C. Bockstal
LAAS-CNRS
Toulouse, France
E-mail: hhassan@laas.fr

KEYWORDS
Differential, Modelling, TCP/IP.

ABSTRACT

Internet is the main communication platform for most multimedia applications of our days. The deployment of new multimedia applications requires precise performance evaluation studies relying on accurate traffic models. Meanwhile, most applications on the Internet are transported by (TCP/IP). Hence, modelling and simulating TCP/IP is an essential tool for performance evaluation and validation studies. Furthermore, simulation techniques and tools must combine accuracy and efficiency when it comes to large scale networks (e.g. Internet). In this paper we present a differential modelling technique applied to TCP/IP. The presented model describes analytically the behaviour of TCP/IP, where fluid differential equations are mixed with control events. Control events pilot the simulation to pass from one differential equation to another. Network nodes are represented by D(t)/D/1/N queues, while D(t) stands for transient deterministic arrival. Losses and delays are evaluated analytically. The model is validated against event-driven simulations of TCP/IP performed in Distributed Hybrid Simulator (DHS) of LAAS-CNRS.

INTRODUCTION

Transmission Control Protocol on IP (TCP/IP) plays an important role in the Internet. Most end-to-end reliable connections on the Internet are established by TCP/IP. In-order delivery of packets, lost packets retransmission and the efficient use of bandwidth are functionalities implemented in TCP/IP, and they are behind its success. However, the numerous functionalities of TCP/IP resulted in a sophisticated algorithm. Hence, from a traffic modelling point of view, the reliability of TCP/IP generates "elastic" traffic because of packet retransmission mechanisms. No simple traffic models could be used to generate TCP/IP traffic unless the TCP/IP loss process is reproduced.

Event-driven technique is widely used to simulate TCP/IP. Unfortunately, the increase in the number of generated events makes it unsuitable for large scale network simulations. Many other techniques to simulate the behaviour of TCP/IP analytically are proposed in the literature (e.g. Altman 2000, Barakat 2000). Most of them are based on analytical stationary approximations of rate and loss process. However, such approaches do not reproduce the transient behaviour of TCP/IP. Our objective is to model TCP/IP analytically to overcome scaling problems while preserving the TCP/IP transient behaviour for more precision. We achieve this using the Differential Traffic Theory (Garcia et al 2001).

In this paper we present a differential model for TCP/IP. The model describes precisely the behaviour of TCP/IP by fluid differential equations mixed with control events. The paper is organized as follows: In section 2 we give an overview of TCP/IP. In section 3 we introduce the differential model of TCP/IP and finally in section 4 we validate our model against event-driven simulations of TCP/IP achieved in DHS (Distributed Hybrid Simulator). We conclude by some remarks and future work.

TCP/IP OVERVIEW

Upon its creation, TCP/IP objective was to control the flow of packets so as not to overload the receiving host. This was achieved by advertising a maximum receiver window upon connection set up (Postel 1981). The early versions of TCP/IP did not consider the network status, a TCP/IP source injected new traffic into the Internet whenever it finds an empty place in the receiver buffer. This has motivated Van Jacobson in 1988 (Jacobson 1988) to introduce his well known congestion control algorithm by a window-based approach. A variable window called congestion window increases when the network is not congested and backs off when congestion occurs. The congestion window is always bounded by the window advertised by the receiver so that TCP/IP continues to do its end-to-end flow control. The evolution of the congestion window is coupled with another value called CREDIT calculated at the source. Indeed, the TCP/IP source sends packets only when it has enough CREDIT. The CREDIT is a calculated by:

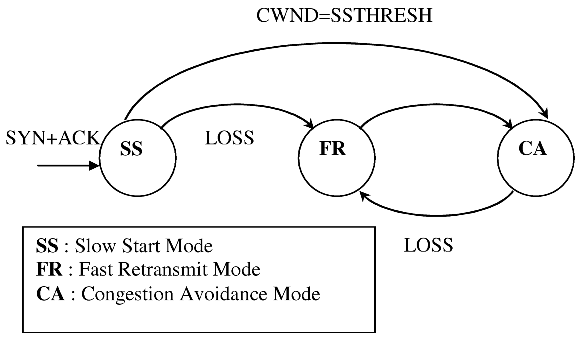
$$CREDIT = ACK + \min(CWND, RWND) - SEQ + NDUP * MSS$$

CWND and RWND are the congestion window and the receiver window respectively. ACK is the address of the last acknowledged packet, SEQ is the address of the last sent packet, and NDUP is the number of duplicate ACKs received by the source and MSS is the maximum segment size.

TCP/IP is characterized by different operation modes. Figure 1 illustrates the transitions between TCP/IP operation modes. We will explain those different modes in the following sections.

Slow Start

When a new connection is established, the congestion window is initialized to one segment. Each time an ACK is received, the congestion window is increased by one segment. The sender starts by transmitting one segment and waiting for its ACK. When that ACK is received, the congestion window is incremented from one to two, and two segments can be sent. When each of those two segments is acknowledged, the congestion window is increased to four and so forth. For each b packets received one ACK is sent from the receiver to the sender. The number of packets sent in one burst follows approximately a geometric series of $(b+1)/b$ reason. This geometric series is the reason behind the exponential behaviour of bit rate during Slow Start mode of TCP/IP. However, the packet rate is controlled by the service rate of the slowest router on the packets path denoted $\mu_{\min}$. ACKs arrive to the sender at this maximum rate and the increase of the Congestion Window CWND is exponential until ACK back rate reaches this value where it becomes linear.



Figures 1: Simplified TCP/IP State Transitions

Congestion Avoidance

Congestion avoidance mode explores the available bandwidth carefully. It is implemented to prevent rapid congestion. Congestion avoidance increments CWND by $(b+1/CWND)$ segment each time an ACK is received. This is a linear growth of CWND, compared to Slow Start's exponential growth. This mode is activated when the value of CWND (during Slow Start) reaches SSTHRESH (a threshold value).

Fast Retransmit

The detection of one loss turns TCP/IP NewReno into the Fast Retransmit mode. There are two indications of a packet loss: a timeout and the reception of three duplicate ACKs. The sender must retransmit all lost packet at the rhythm of one packet by RTT. This mode turns over when all lost packets are retransmitted. A Fast Recovery mode is coupled with the Fast Retransmit mode to enhance the performance of TCP/IP. After the TCP/IP sender finishes retransmitting all lost packets, it does not resume in the Slow Start mode. It turns into the congestion avoidance mode instead with a CWND half its value before loss detection.

THE DIFFERENTIAL MODEL OF TCP/IP

The differential model propagates the rate values from one node to another as a function of time. The integration of differential equations is done each Δt (time step) which is considered constant. We assume that all values calculated with a function f verify the following equation:

$$f(t+\Delta t) = f(t) + \dot{f}(t) * \Delta t \qquad (1)$$

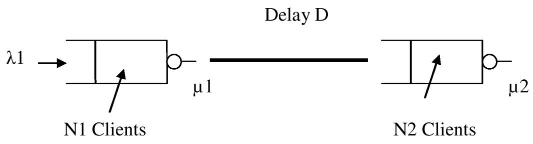
While:

$$\dot{f}(t) = K, \forall t \in [t, t+\Delta t] \qquad (2)$$

K is a constant. That means we consider the variation of function f is null during the interval $[t+\Delta t]$.

The propagation of rates in a network must take into consideration the latency D that exists between different servers due to link delays and server waiting times.

On Figure 2 we represent two server queues with a constant link delay D . The service rate of server1 is $\mu 1$ and the service rate of server2 is $\mu 2$.



Figures 2: Propagation of rates

The source input rate in server1 is $\lambda 1$. $N1$ and $N2$ are the number of packets in queue1 and queue2 (server1 and server2 respectively).

The number of packets in queue1 evolves according to equation:

$$N1(t+\Delta t) = N1(t) + (\lambda 1 - \mu 1) * \Delta t \qquad (3)$$

Server1 can serve $\mu 1 * \Delta t$ data quantity during the integration step. The output rate of server1 is ruled by:

$$\overset{*}{\mu}1 = \begin{cases} \mu 1 & if \quad N1(t+\Delta t) \geq 0 \\ \lambda 1 & if \quad N1(t+\Delta t) < 0 \end{cases} \qquad (4)$$

The output rate of queue1 constitutes the input rate of queue2. The propagation of $\overset{*}{\mu}1$ to queue2 must be done respecting the latency of the link between two servers. The input rate of queue2 is given by the equation:

$$\lambda 2(t) = \overset{*}{\mu}1(t-D) \qquad (5)$$

Slow Start

Let $RTT(t)$ be the Round Trip Time at date t. TCP/IP transmit packets by bursts according to its CREDIT value.

Let tb_{i+1} bet the date of transmission of burst *i+1* and tb_i the date of transmission of burst *i*. We have directly the following equation:

$$tb_{i+1} = tb_i + RTT(tb_i) \quad (6)$$

The source receives one ACK for each b packets. The credit increase by one for each received ACK during Slow Start. The rate of TCP/IP during Slow Start follows a geometric series of $(b+1)/b$ reason. Let $\lambda(t)$ be the rate of TCP/IP source. We get:

$$\lambda((k+1)\cdot RTT) = \frac{b+1}{b}\cdot \lambda(k\cdot RTT) \quad (7)$$

This could be written as:

$$\lambda_k = \lambda_0 \left(\frac{b+1}{b}\right)^{\frac{k}{RTT}} \quad (8)$$

Notice that the number of sent packets at the beginning of transmission is b packet, λ_0 can be derived easily:

$$\lambda_0 = \frac{b}{RTT}$$

The derivative form of equation (8) is:

$$\dot{\lambda}(t) = \frac{\lambda(t)}{RTT(t)}.\ln\left(\frac{b+1}{b}\right) \quad (9)$$

Equation (9) is general and depends only on the value of b (constant equals to 2 generally) and the estimated value of *RTT(t)* at time t.
The TCP/IP source rate is controlled by ACK back rate. In Slow start mode for each ACK received the source can transmit $b+1$ packets. The maximum rate value $\lambda_{r-\max}$ is given by:

$$\lambda_{r-\max} = (b+1).\lambda_{ack} \quad (10)$$

The value of λ_{ack} could be expressed as a function of the lowest service rate $\mu_{\min}$ on the packets path. We have $\lambda_{ack} = \frac{\mu_{\min}}{b}$ as we receive one ACK for each b packets, and $\lambda_{r-\max}$ become:

$$\lambda_{r-\max} = \mu_{\min}\frac{b+1}{b} \quad (11)$$

The ACK back rate is not the only limitation on the TCP/IP source rate. In fact the TCP/IP source discovers the available bandwidth by increasing its congestion window. The congestion window CWND is limited by a maximum value. That means the TCP/IP rate increases and stabilizes on the following value:

$$\lambda_{cwnd-\max} = \frac{CWND_{\max}}{RTT(t)} \quad (12)$$

Congestion Avoidance

During Congestion Avoidance, TCP/IP continues to explore the available bandwidth, trying to get the maximum while avoiding quick congestion. The TCP/IP average rate is estimated by:

$$\lambda(t) = \frac{CWND(t)}{RTT(t)} \quad (13)$$

The CWND evolution is less important than in Slow Start mode. We will see the evolution of CWND later. Here again the rate is bounded by the minimum server service rate $\mu_{\min}$. We have:

$$\lambda_{r-\max} = (b + \frac{1}{CWND(t)}).\lambda_{ack} \quad (14)$$

Recall that during congestion avoidance the source can transmit only $b + \frac{1}{CWND(t)}$ for each received ACK. By replacing $\lambda_{ack} = \frac{\mu_{\min}}{b}$ we get:

$$\lambda_{r-\max} = (b + \frac{1}{CWND(t)}).\lambda_{ack} \quad (15)$$

The limitation on the maximum CWND value is also valid in congestion avoidance. Equation (12) applies on TCP/IP rate during Congestion Avoidance.

Fast Retransmit

After the detection of a loss the source passes in Fast Retransmit mode. In this mode packets lost are sent one by one for each ACK received (or one packet by *RTT(t)*). The rate formula is:

$$\lambda(t) = \frac{1}{RTT(t)} \quad (16)$$

When all lost packets are sent, the source gets into Congestion Avoidance mode with half the value of CWND before loss detection.

CWND Evolution

TCP/IP is a window controlled algorithm. The evolution of the congestion window is resumed here:

$$CWND(t) = \left\{\begin{matrix} CWND(t)+1 & SS \\ CWND(t)+\frac{1}{CWND(t)} & CA \\ 0 & FR \\ \frac{CWND(t)}{2} & Loss \end{matrix}\right\} \quad (17)$$

ACK Back Rate

The ACK back rate represents the rate at which ACK packets reaches to the source. The value of this rate determines the evolution of the Congestion Window and by consequence the TCP/IP source rate. Let $\lambda_j(t)$ be the ACK back rate and $\lambda_i(t)$ is the TCP/IP source rate, we have the following relation:

$$\lambda_j(t) = \begin{Bmatrix} \lambda_i(t) & FR \\ \dfrac{\lambda_i(t)}{b} & Otherwise \end{Bmatrix} \tag{18}$$

Node Modelling

Let $\lambda_i(t)$ be the input rate of node *i* and $\mu_i(t)$ its output rate. The number of packets in queue is denoted $N_i(t)$.

We calculate the output rate $\overset{*}{\mu}_i(t)$ of node *i* at date t as follows:

$$\overset{*}{\mu}_i(t) = \begin{cases} \lambda_i(t) & if \quad (\lambda_i(t) < \mu_i(t)) \,\&\, (N_i(t) \neq 0) \\ \mu_i(t) & if \quad (\lambda_i(t) \geq \mu_i(t)) \,\&\, (N_i(t) \neq 0) \\ 0 & if \quad N_i(t) = 0 \end{cases} \tag{19}$$

A buffer overflow fires the loss detection event. In TCP/IP NewReno the loss detection is achieved by triple ACK reception at source. The delay between the moment the loss occurs and the moment the source is informed about it is equivalent to one RTT plus three packet service time delay:

$$\Delta t3 = RTT(t) + \frac{3}{\mu_{\min}} \tag{20}$$

During this time all packets transmitted by the source before detecting the first packet loss are also lost.

RTT Estimation

The RTT value depends on the path followed by packets. Let D_i be the delay of Link *i* between router *i* and router *i+1*, and T_i the processing delay of one packet in node *i*. RTT is estimated by:

$$RTT(t) = \sum_{i \in R} T_i(t) + D_i \tag{21}$$

With *R* the group of Routers and:

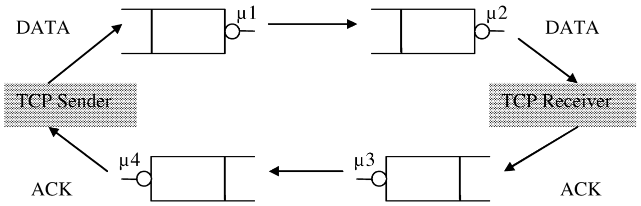
$$T_i(t) = \begin{Bmatrix} \dfrac{1}{\mu_{\min}} & N(t) = 0 \\ \dfrac{N(t)}{\mu_{\min}} & N(t) > 0 \end{Bmatrix} \tag{22}$$

While $N(t)$ represents the load (in packets) of the queue at date *t*.

SIMULATIONS

Simulation Network

We validate our model using the network described on Figure 3. This network is very representative and allows us to evaluate the different parameters of our model.



Figures 3: Simulation Network

We will show the performance of our model in each of the TCP/IP operation modes. Various network parameters will be varied to test the robustness of the model.

Validation Tests

Case Study Test

All curves shown in this section are obtained by the simulation of the network presented on Figure 3 with the simulation parameters listed in Table 1.

Table 1: Simulation Parameters

µ (Bytes/s)	Buffer (packets)	Link delay (ms)
500000	35	1

On Figure 4 we show the evolution of the congestion window via differential and event-driven simulations. The curves show the good behaviour of differential simulation compared to event-driven simulation for all operation modes.

As the instantaneous rate is difficult to obtain in event-driven simulations we compare the number of sent packets in both cases (see Figure 5).

We observe a little difference between the differential model and the event-driven simulation. The Fast Retransmit phase lasts less than expected. In fact, we detect one lost packet less because of the fluid approximation of rates. As a consequence the retransmission period of lost packets lasts less than in event-driven simulation. This difference is of one RTT the time of one lost packet retransmission.

Globally we obtain very good results. Packets rate and losses number are evaluated precisely. In Table 2 we give the global statistics of our simulation.

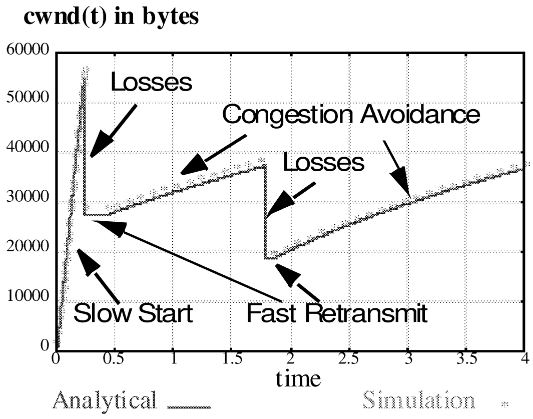


Figure 4: Evolution of CWND(t) Analytical vs Event-Driven

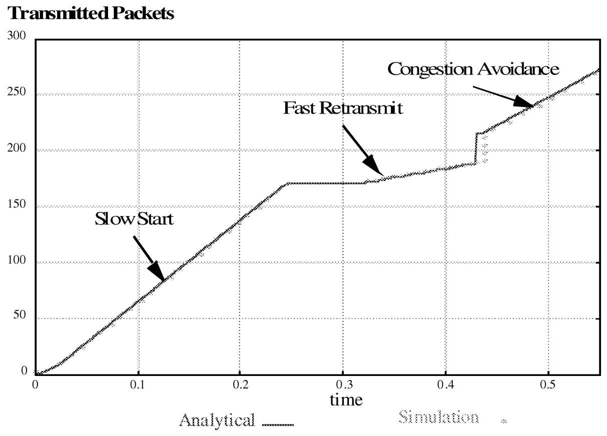


Figure 5: Number of Transmitted Packets By The Source

Table 2: Global statistics

	Analytical	Simulation	Relative Error
Rate (Packets/s)	488.7	488.14	0.11%
Loss ratio	0.266%	0.276%	3.831%

Global Validation
The results showed in previous section point out clearly the precision of the differential model. We will present now summarized results on a large number of simulation configurations.
We use the same network of Figure 3. We vary the value of service rate (μ) and buffer capacity (buffer). For each couple {μ, buffer} we vary the size of transferred file. The values are resumed on Table 3.
We perform 400 simulations using these configuration parameters. For each configuration couple {μ, buffer} we evaluate the transmission duration, the source packet rate, the input rate and the output rate (ACK) of the receiver as well as the loss ratio of the connection. For each file size we calculate the relative error for each of the mentioned values comparing with the event-driven simulation. We calculate the average of these relative errors by couple {μ, buffer}. Finally, we show the global average of these simulations, resulting by averaging all obtained averages by couple {μ, buffer}. We give also the standard deviation.

Table 3: Global Configuration Parameters

μ	{1, 2, 5, 8 , 10, 20, 50, 70, 100}*16384 Bytes/s
Buffer	{10, 20, 30, 40, 50} packet
File size	{1, 2, 3, 5, 8, 10, 20, 50, 100, 200, 500}*102336 Bytes

Table 4: Relative error (%) with link delay 0.001s

Link delay 0.001 s	Trans duration	Source rate	Recep rate	ACK rate	Losses
Mean (%)	1.68	1.63	1.63	1.9	7.48
Standard deviation (%)	0.89	0.76	0.89	1.07	4.87

The global validation shows that the differential model works very well.

CONCLUSION

In this paper, we presented a differential analytical model of one TCP/IP connection. Our model is based on differential equations describing the rate variation using D(t)/D/1/N network node model. The model evaluates rates and losses analytically. We obtained very good results compared to event-driven simulations.
We are working on the extension of our model to consider multiple connections in a network environment. In multi connection environment the detection of losses is more sophisticated as the buffer overflow is due to an aggregated flow. A sharing mechanism of the node output rate needs to be implemented.

REFERENCES

Altman E., K. Avrachenkov, C. Barakat, "A Stochastic Model for TCP/IP with Stationary Random Losses", *ACM SIGCOMM*, Aug. 2000.
Barakat C., E. Altman, "A Markovian model for TCP Analysis in a Differentiated Services Network", *(QofIS)*, Sep. 2000.
Braden R., "Requirements for Internet Hosts - Communication Layers" RFC 1122, Oct. 1989.
Garcia JM., D. Gauchard, O. Brun, P. Bacquet, J. Sexton, E. Lawless, "Modélisation différentielle du trafic et simulation hybride distribuée" *Performances des réseaux et systèmes*, Vol.13, N°6, pp.635-664, 2001.
Jacobson V., "Congestion Avoidance and Control" *ACM SIGCOMM*, Aug. 1988.
Postel J., "Transmission Control Protocol" RFC 793, Sep. 1981.

TRANSIENT ANALYSIS OF SEMI-MARKOVIAN SWITCHING SYSTEMS IN TELECOMMUNICATION NETWORKS

Gerhard Hasslinger
T-Systems, Technologiezentrum
Deutsche Telekom Allee 7, D-64295 Darmstadt,
Germany, E-mail: gerhard.hasslinger@t-systems.com

Sebastian Kempken
Abteilung Informatik
Universität Duisburg-Essen, D-47048 Duisburg,
Germany, E-mail: kempken@inf.uni-due.de

KEYWORDS

Discrete time server systems; transient and steady state analysis; semi-Markov processes; busy periods.

ABSTRACT

The transient behaviour of single server systems is investigated to demonstrate that transient analysis provides a simple and flexible method especially in discrete systems. We consider a basic time slotted model for telecommunication systems, using semi-Markovian processes to characterize highly variable traffic as often measured on Internet platforms. Evaluations of the distribution of the workload and waiting times are based on transition equations starting from some predefined initial situation. In this way, the convergence to steady state situations can be examined and the system performance can be evaluated from different views, e.g. by analysing the length of busy and overload periods of the server.

1 INTRODUCTION

Transient analysis of stochastic systems is a basic method which is related to steady state analysis as well as simulation. Matrix analytic methods [2][3][8] and factorisation approaches [6][7][9][14] provide efficient computation schemes for steady state solutions of systems with regular structures in the transition equations. In addition, transient analysis shows the evolution of the system from a predefined starting situation over a period of time or in long term development. When the system is ergodic, then convergence to a steady system state is observed over time. Thus, transient system analysis may also be used for the steady state as a limiting behaviour, but the computational effort is often essentially higher than for direct steady state solutions depending on the convergence properties and on the relevant state space of the system. On the other hand, the transient analysis is more flexible in order to include exceptional cases and non-regular structures in the system transition matrix.

In comparison to simulation, transient analysis provides complete distribution functions of system states at embedded points in time, whereas simulation follows randomly chosen paths of system development yielding results subject to statistical deviations at a confidence level. Thus transient analysis attracts much attention for different applications especially in telecommunication [1][4][10][11].

State space limitation is an important precondition to make transient analysis feasible. If the state space becomes too large, only those states with probability $>10^{-k}$ may be considered for an approximate transient analysis in a feasible state space. This generates a tendency to extract a set of states surrounding the maximum likelihood behaviour of the system [15]. On the other hand, it restricts the ability to indicate regions of very small probability included in transient distributions of the complete state space. Equilibrium point analysis [10][13] is another method to estimate the relevant points of stable system behaviour, which may be complemented by transient analysis to get better insight into their neighbourhood.

In the next section we introduce semi-Markovian systems in discrete time and derive their basic transition equations in section 3. The evaluation of the workload and waiting times in the system is demonstrated in a extensive example in section 4, which is extended to include the busy periods in section 5.

2 SEMI-MARKOV MODEL OF SWITCHING SYSTEMS IN TELECOMMUNICATION

Switches or routers in telecommunication networks accept data from multiple input links and distribute them over a number of output links in the direction of their destination. The output line is chosen by IP routers for each packet according to forwarding principles of routing protocols, e.g. the OSPF and BGP routing standards of the Internet engineering task force (IETF). In addition, buffers are included. Usually there is a buffer in front of each output, which is able to collect all data packets destined for the output and to store them in case of collisions and temporary overload before they are forwarded after some delay.

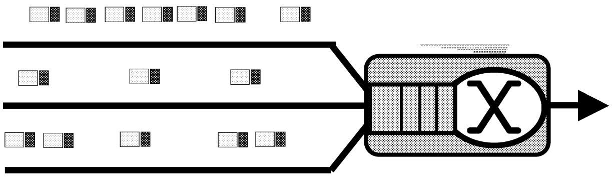


Figure 1: A buffered multiplexing or switching system

Thus a usual router or switch model considers

- an output link with a forwarding capacity C measured e.g. in Mb/s (Megabit/s) or Gb/s (Gigabit/s),
- a buffer of size B measured e.g. in Mb or Gb,
- the input process of all those packets arriving at the switch, which are destined to and forwarded by the considered output link.

In order to characterize the input and forwarding process, we introduce a time slotted system, as often utilized in performance analysis of telecommunication systems [9] [5][6]. Therefore the time is divided into slots of equal length Δ and the amount of arriving and forwarded data is traced per slot. Usually a constant amount $C\Delta$ of data can be forwarded per slot by the switching system.

On the other hand, the arrival processes in telecommunication systems are often highly variable. Semi-Markov processes are an appropriate approach, which includes many types of variability and still can be analysed efficiently.

3 DEFINITION OF THE MODEL

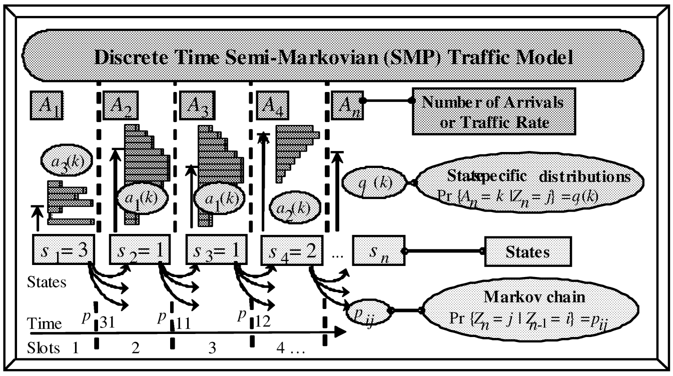

Discrete Time Semi-Markovian (SMP) Traffic Model
A_1
A_2
A_3
A_4
A_n
Number of Arrivals or Traffic Rate
$a_3(k)$
$a_1(k)$
$a_1(k)$
$a_2(k)$
$q(k)$
Statespecific distributions
Pr {$A_n = k$ | $Z_n = j$} $= q(k)$
$s_1 = 3$
$s_2 = 1$
$s_3 = 1$
$s_4 = 2$
...
s_n
States
States
Markov chain
Pr {$Z_n = j$ | $Z_{n-1} = i$} $= p_{ij}$
p_{ij}
Time
p_{31}
p_{11}
p_{12}
Slots 1 2 3 4 …

Figure 2: Semi-Markovian state model for arrivals per time slot

We consider the workload of a time-slotted semi-Markov server. Let s_t denote the current state of the system in slot t and w_t denote the workload at the beginning of slot t. Then

$$W_t = (w_t, s_t) \in \mathbb{N}_0 \text{ x } \{1, \ldots, M\} \quad \text{for } t \in \mathbb{N}_0,$$

includes all information about the current state of the discrete time semi-Markov server, which is relevant for future development of the workload.

In particular, changes of the workload from one slot to the next will follow state specific distribution functions for the arrival and service process. Let A_t and S_t denote random variables for the number of new arrivals and for the service capacity during time slot t. We assume that service is done at the end of a slot. Then $U_t = A_t - S_t$ determines the increase or decrease of the workload per slot via Lindley's equation:

$$w_{t+1} = \max(w_t + A_t - S_t, 0) = \max(w_t + U_t, 0).$$

To make the computation tractable, we assume that A_t and S_t are bounded, where $0 \le A_t \le h$; $0 \le S_t \le g$ and thus $-g \le U_t \le h$. For application to switches and routers in telecommunication, the amount of arriving data per slot as well as the capacity of the forwarding process serving the data is always bounded by the sum of capacities of the input lines from which the arriving data is delivered and by the capacity C of the output line. In particular, assuming a fixed slot length Δ, the forwarding capacity of the switch or router is constantly at $C\Delta$, representing a deterministic service.

In general, we include non-deterministic and often highly variable semi-Markovian arrival and service processes as illustrated in Figure 2:

$$\begin{aligned} & a_i(k) = \Pr(A_t = k \mid s_t = i) \quad \text{for } 0 \le k \le h;\ 1 \le i \le M \\ & s_i(k) = \Pr(S_t = k \mid s_t = i) \quad \text{for } 0 \le k \le g \\ \Rightarrow \quad & u_j(k) = \Pr(U_t = k \mid s_t = i) \quad \text{for } -g \le k \le h. \end{aligned}$$

$a_i(k)$ denotes the probabilities that k units arrive in a slot, while the system is in state i in the current slot. $s_i(k)$ and $u_i(k)$ are corresponding notations for the service capacity and the difference between the amount of arriving and served units. In addition, the state s_t may change from a slot to the next one according to an underlying Markov chain with transition probabilities p_{ij}:

$$\Pr(s_{t+1} = j \mid s_t = i) = p_{ij}.$$

We use a common state space for the arrival and service process. Even if we start from two distinct and independent underlying Markov chains, one of them governing the arrival and the other the service process, both can be combined into a single chain via Kronecker's product. We assume that the underlying Markov chain is irreducible and that steady state probabilities p_i exist, which are given by a homogeneous system of linear equations together with a normalization constraint:

$$p_i = \Sigma_j\ p_j p_{ji} \quad \text{for } i = 1, \ldots, M \text{ and } \Sigma_j\ p_j = 1.$$

Favourable properties of semi-Markov processes are simple expressions that can be straightforwardly derived for their n-step transitions and their autocorrelation function [5].

According to the previously defined behaviour of a semi-Markov server, we obtain the transition probabilities of the workload W_t until the next slot W_{t+1}:

$$\Pr(W_{t+1} = (l, j) \mid W_t = (k, i)) = u_i(l-k)\, p_{ij} \qquad (1)$$

for $0 < l < N$, where $u_i(m) = 0$ for $m < -g$ and $m > h$. We assume that the buffer size and therefore also the workload at the server is limited to N, such that arrivals exceeding the limit are rejected or dropped. In this case, not the complete amount of data arriving in the affected slot is dropped, but only the amount exceeding the buffer size N. This leads to special transition equations for the boundary states of the workload at 0 and N:

$$\begin{aligned} \Pr(W_{t+1} = (0, j) \mid W_t = (k,i)) &= \textstyle\sum_{n=-g}^{-k} u_i(n)\, p_{ij}; \\ \Pr(W_{t+1} = (N, j) \mid W_t = (k,i)) &= \textstyle\sum_{n=N-k}^{h} u_i(n)\, p_{ij}. \end{aligned} \qquad (2)$$

A solution of the stationary workload distribution W_t of semi-Markovian servers is presented in [14]. In this paper we focus on the transient behaviour of the system. Therefore we assume starting conditions, which may be given by an initial state or, in general, an initial distribution for the state in the first considered slot.

Then the previous equations (1) and (2) to compute the workload and state probabilities of W_{t+1} from known W_t are sufficient to determine the development of the workload in the next and all the following slots. The computational complexity for a step in the transient process to get from W_t to W_{t+1} is bounded in the order $O((g+h)M^2N)$.

As a stability condition for a system with unlimited buffer, the mean service time is assumed to be smaller than the mean interarrival time

$$E(S) = \Sigma_j \Sigma_k\, p_j \cdot k \cdot s_j(k) < E(A) = \Sigma_j \Sigma_k\, p_j \cdot k \cdot a_j(k) \Leftrightarrow E(U) < 0.$$

In the sequel we always evaluate systems with limited buffer size and thus limited state space. A stable system with infinite buffer may be approached by system analysis with limited buffer by stepwise increasing the buffer size until the probabilities of buffer overflow are become negligibly small.

The service discipline is non pre-emptive and the order of service is independent of the service time, e.g. first come first served.

4 EXAMPLE

We illustrate basic results derived from the transient analysis in an example of a semi-Markovian server with application to switching systems in the Internet. We assume that the output line has a capacity of 10Gb/s as a usual STM-64 wire speed of Internet backbones. Time slots are considered on the millisecond time scale Δ = 1ms, such that the switch can forward 1Mb of data per slot.

The traffic arriving at the output is modelled by an SMP with 4 states. The states are valid per slot and can change in the next slot according to the following transition matrix:

$$P = (p_{ij}) = \begin{pmatrix} 0.7 & 0.3 & 0 & 0 \\ 0.3 & 0.5 & 0.2 & 0 \\ 0 & 0.2 & 0.5 & 0.3 \\ 0 & 0 & 0.3 & 0.7 \end{pmatrix}$$

Incoming traffic is characterized by state dependent distributions. We assume that the traffic amount per slot has a uniform distribution with different ranges per state:

Range for state 1: 0.7 – 0.9 Mb,

range for state 2: 0.8 – 1.0 Mb,

range for state 3: 0.9 – 1.1 Mb and

range for state 4: 1.0 – 1.2 Mb.

Accordingly, the traffic rate increases with the state number, ranging from 70 – 90 % load in the first state up to an overload case of 100 – 120 % load in state 4. For a discrete time modelling, we adopt step distribution in each case with 20 steps. The choice of 20 steps is a compromise between accuracy and tractability demands. In general, we can use more steps for a closer adaptation to a continuous uniform distribution at the expense of increasing computation effort.

Thus in the example we assume

$a_1(71) = a_1(72) = \ldots = a_1(90) = 0.05;$

$a_2(81) = a_2(82) = \ldots = a_2(100) = 0.05;$

$a_3(91) = a_3(92) = \ldots = a_3(110) = 0.05;$

$a_4(101) = a_4(102) = \ldots = a_4(120) = 0.05$

as state specific distributions where $a_i(k)$ = 0 for all values which are not mentioned. This implies a step unit corresponding to 0.01 Mb, which is also valid for the workload distribution, i.e. the units of the workload are also at the size of 10 kb.

Since the switch is capable of forwarding 100 units per time slot, the state specific distributions of the differences in the workload observable per slot are given by:

$u_1(-29) = u_1(-28) = \ldots = u_1(-10) = 0.05;$

$u_2(-19) = u_2(-18) = \ldots = u_2(0) = 0.05;$

$u_3(-9) = u_3(-8) = \ldots = u_3(10) = 0.05;$

$u_4(1) = u_4(2) = \ldots = u_4(20) = 0.05.$

Results of the transient analysis of the system are depicted in the following graphs. Our starting point is always at an empty system in state 4: W_0 = (0, 4). The buffer of the system is limited at 1000 units or 10Mb.

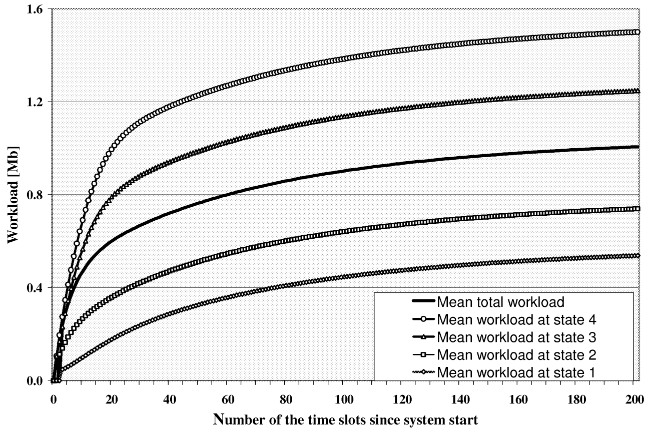


Figure 3: Development of the mean workload of the server

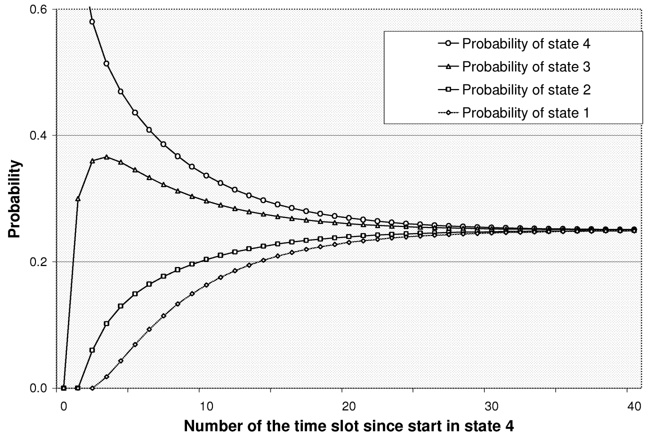


Figure 4: Probability to enter state k in a slot, starting from k= 4

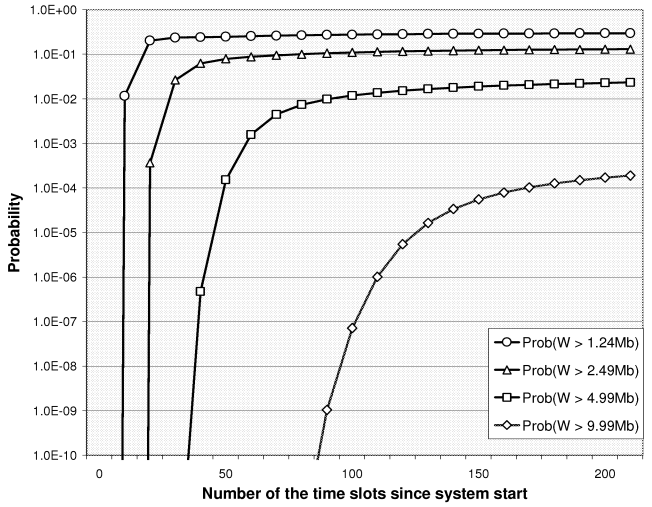


Figure 5: Probability to exceed workload levels after n slots

The mean workload, which builds up in the next 200 slots is shown in Figure 3 in total and also depending on the state being valid in the slot. Figure 4 shows the convergence to a uniform steady state distribution in the underlying Markov chain.

The initial state 4 is also encountered with high probability in the next slots, but after 40 slots all 4 states are observed at almost the same probability. Figure 5 shows the probabilities that the workload exceeds certain levels. Initially, the workload is zero and it lasts for a while until a considered level is reached at all. The probabilities for exceeding a workload threshold are increasing until steady state conditions are approached. As can be observed, convergence to steady state is faster in the underlying chain than for the workload.

The steady state distributions can be directly determined and confirmed by factorisation methods or matrix analytical methods. We have implemented the Wiener-Hopf [6] and the polynomial [14] factorisation for classical SMP/D/1 queueing systems. Both approaches are directly transferable to time slotted systems including the considered example, since Lindley's equation applies for classical queueing systems in the same way as for time slotted semi-Markovian systems, i.e. we have to construct a SMP/D/1 system with the same distribution of the workload differences $u_j(k)$ and transition matrix p_{ij}. Then the workload development of the SMP/D/1 system immediately prior to arrivals is equivalent to the workload process in the time slotted model, although the arrival and service processes have a different interpretation in both cases.

5 ANALYSIS OF BUSY PERIODS

The previous analysis of the transition equations can be modified to obtain the distribution of the busy period length. Therefore we start at the beginning of a busy period in state s of the underlying Markov chain, i.e.

$$\Pr(W_0 = (0, s)) = 1.$$

For the following slots we compute a defective distribution of the workload $\tilde{W}_t(s)$, restricted to the case that the busy period starting in slot 0 at state s is still ongoing until the current slot. We assume that a busy period ends as soon as the server is able to completely forward the workload in a slot, i.e. when a state $(0, r)$ is reached. Then r is the initial state of the next busy period. Thus we obtain $\tilde{W}_0(s) = W_0 = (0, s)$ and for $t > 0$:

$$\begin{aligned}&\Pr(\tilde{W}_t(s) = (k,i)) =\\ &\Pr(W_t = (k,i) \text{ and } \forall\, j \le t, v \in (1,...,M) : W_j \ne (0,v)).\end{aligned}$$

We simply can compute the defective distribution of $\tilde{W}_t(s)$ by using equation (1) in the same way as for the distribution of W_t and modifying the boundary condition for the states $(0, s)$ in equation (2) into

$$\Pr(\tilde{W}_{t+1}(s) = (0, j) \,\big|\, \tilde{W}_t(s) = (k,i)) = 0. \qquad (3)$$

The modified equation (3) drops all probability mass which would correspond to entering a new busy period at slot t +1. Therefore the remaining probability mass in the calculation of $\tilde{W}_t(s)$ is restricted to the case that the workload has been always positive since the system start. Again, we compute the distribution of $\tilde{W}_t(s)$ by a transient analysis for t = 1,2, 3, … based on equations (1), (3) and the second equation of (2).

In addition, we obtain the probability mass for entering a new busy period after t slots, separated for each subsequent state r at the beginning of the next busy period. Let $b_{(s\to r)}(t)$ denote the probability that a busy period starting in state s is finished after t slots at state r as initial state of the next busy period. Then we obtain:

$$b_{(s\to r)}(t) = \sum\nolimits_{i=1}^{M} \sum\nolimits_{k>0} \Pr(\tilde{W}_{t-1}(s) = (k,i)) \sum\nolimits_{j=-k}^{-g} u_i(j) \cdot p_{ir}. \quad (4)$$

Finally the distribution function $B_s(m)$ of a busy period with duration m starting in state s is given by:

$$\begin{aligned}B_s(m) &= 1 - \sum\nolimits_r \sum\nolimits_{t<m} b_{(s\to r)}(t)\\ &= \sum\nolimits_{k>0} \sum\nolimits_{i=1}^{M} \Pr(\tilde{W}_m(s) = (k,i)).\end{aligned} \qquad (5)$$

Results for the length of busy periods are shown in Figure 6. On the right hand, the same case is considered that has been analysed previously in section 4. The figure on the left shows the analysis for a busy period starting in state 3. In state 1 and 2 a busy period ends immediately, since the state specific workload differences are always negative or zero in those cases. The distribution of the probability for busy period lengths larger than n slots is plotted in Figure 7. While most busy periods are short, there is also a non-negligible fraction of very long busy periods. This also has been experienced in studies of busy periods of the classical GI/G/1 queueing systems [6].

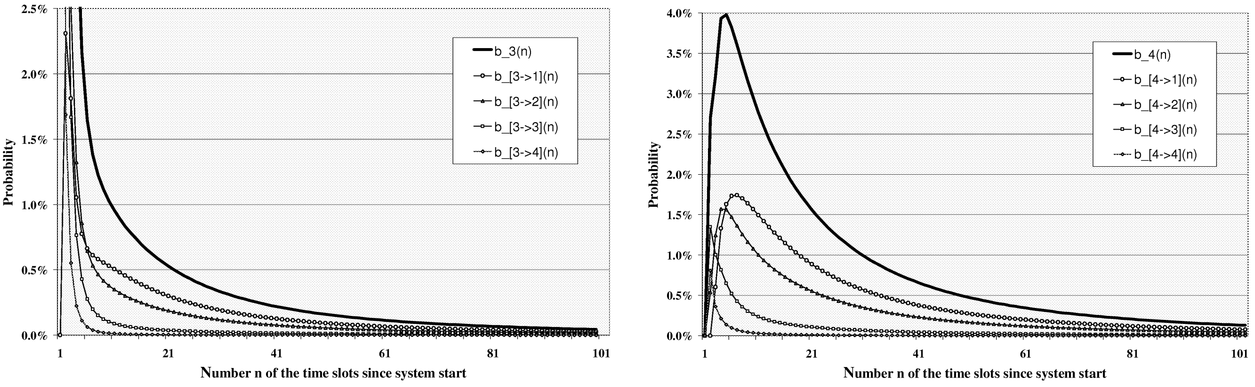


Figure 6: Busy period length: Probabilities that a busy period lasts for k slots depending on the initial state s and the state r afterwards $b_{(s\to r)}(k)$ and $b_s(k) = \Sigma_r \Pr(B_{(s\to r)}(k))$.

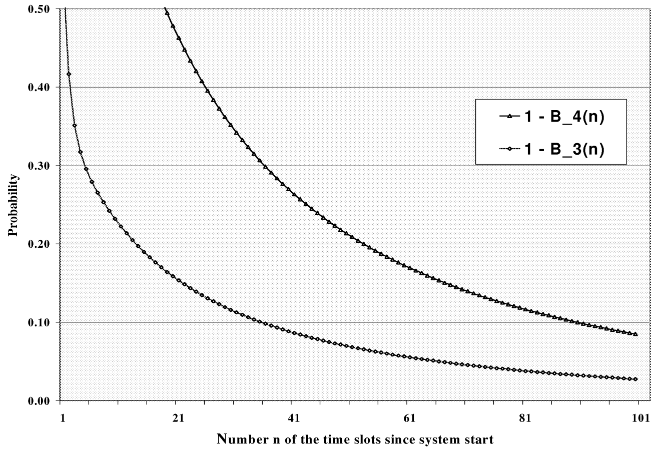


Figure 7: Probabilities of a busy period exceeding length *n* for initialisation in state 3 ($B_3(n)$) and in state 4 ($B_4(n)$)

6 CONCLUSION

Transient analysis can be used to obtain manifold properties of server systems including

- the development of the system over time from arbitrary starting conditions, e.g. a system state or an initial distribution, converging to steady state behaviour, if conditions for a stable or stationary system are valid,
- characterization of the departure process by analysing the distribution of the length of a busy period and the following idle period,
- first passage times, i.e. the time until a predefined set of states is reached for the first time etc.

The underlying transition equations of the system can be flexibly adapted to modified boundary conditions for systems with finite waiting room and many other special cases. Challenges of the method can be seen in the computational complexity and numerical accuracy. A compact representation of the state space is mandatory to keep the analysis tractable for a wide range of server systems. Investigations by [14] also show how the accuracy level can be evaluated by verified computation techniques based on interval arithmetic enclosing the exact result or by sensitivity analysis with regard to input parameters.

If the state space becomes too large, transitions may be followed only for a limited set of transient states, e.g. those with probabilities above a minimum threshold or some other criterion for their relevance. This allows an approximate analysis of a wider spectrum of systems.

This work is a step in the development of a tool for verified analysis of transient and steady states of server and queueing systems [7] by integrating the discussed potential of transient analysis with steady state analysis and verification methods.

REFERENCES

[1] S. Ahn and V. Ramaswami, Efficient algorithms for transient analysis of stochastic fluid flow models, J. Appl. Probab. 42/2 (2005) 531–549

[2] A.S Alfa, Combined ellapsed time and matrix-analytic method for the discrete time GI/GI/1 and GI^X/G/1 system, Queueing systems, 45/1 (2003) 5-25

[3] A.S Alfa and W. Li, Matrix-geometric analysis of the discrete time GI/G/1, Stochastic Models, 17/4 (2001) 541-554

[4] L. Breuer, Numerical results for the transient distribution of the GI/G/1 queue in discrete time, Proc. Measurement, Modelling and Eval. of Computer and Commun. Syst. (MMB 13) Nuremberg (2006) 209-218

[5] G. Hasslinger, Semi-Markovian modeling and performance analysis of variable rate traffic, Telecommunication Systems 7 (1997) 281-298

[6] G. Hasslinger, Waiting times, busy periods & output models of a server analysed via Wiener-Hopf factorization, Performance Evalvaluation 40 (2000) 3-26

[7] S. Kempken, W. Luther and G. Haßlinger, A tool for verified analysis of transient and steady states of queues, accepted paper on the Value Tools workshop, Pisa, Italy, 10.-13.10.2006 <www.valuetools.org>

[8] G. Latouche and V. Ramasvami, Introduction to matrix analytic methods in stochastic modeling, Philadelphia, ASA-SIAM, 1999

[9] S.-Q. Li, A general solution technique for discrete queueing analysis of multi-midia traffic on ATM, IEEE Trans. on Commun. COM-39 (1991) 1115-1132

[10] A. Pantazi and T. Antonakopoulos, Equilibrium point analysis for binary exponential backoff, Computer Communication 24 (2001) 1759-1768

[11] S. Rank and H.-P. Schwefel, Transient analysis of buffer-occupancy fluctuations and relevant time scales, Perf. Eval. 63 (2006) 725-742

[12] H.-P. Schwefel, L. Lipsky and M. Jobmann, On the neccessity of transient performance analysis in telecommunication networks, Proc. International Teletraffic Congress (2001)

[13] S. Tasaka, Performance analysis of multiple access protocol by equilibrium point analysis, The MIT Press, Boston (1986)

[14] D. Traczinski, W. Luther and G. Hasslinger, Polynomial factorization for servers with SMP workload: Performance and numerical aspects of a verified solution technique, Stochastic Models 21 (2005) 643-668

[15] P. Whittle, Systems in stochastic equilibrium, John Wiley (1986)

HIGH PERFORMANCE COMPUTING

PERFORMANCE ANALYSIS FOR HIGH-PRECISION INTERCONNECT SIMULATION

R. Heinzl$^\triangle$, M. Spevak$^\circ$, P. Schwaha$^\triangle$, T. Grasser$^\triangle$ and S. Selberherr$^\circ$

$^\triangle$Christian Doppler Laboratory for TCAD in Microelectronics
at the Institute for Microelectronics

$^\circ$Institute for Microelectronics, Technical University Vienna,
Gußhausstraße 27-29/E360, A-1040 Vienna, Austria
E-mail: {heinzl|schwaha|spevak|grasser|selberherr}@iue.tuwien.ac.at

KEYWORDS
Mesh generation, error estimation, mesh quality, high performance computing, programming paradigms

ABSTRACT

This work analyzes the performance of high-precision interconnect simulation tools on refined meshes with guaranted accuracy. On the one hand, the integrated circuits are subject to an ongoing miniaturization which results in ever increasing computing power. On the other hand, the simulation of these integrated circuits demands more sophisticated simulation methodologies such as better resolution of geometrical features or more complex surface topography. We show that modern microprocessor architectures and memory hierarchies impose performance limits on the simulation time.

INTRODUCTION

Down-scaling of integrated circuits to the deep sub-micron regime and beyond increases the influence of interconnects on circuit behavior drastically. Parasitic effects are becoming more and more important as devices get faster and line widths smaller. These effects become the limiting factor for further improvements of circuit speed. An essential step in technology computer aided design (TCAD) is the optimization of these parameters, which demands vast amounts of computer resources, CPU-time and memory. Therefore the performance of current computer systems is essential for an optimal simulation flow. The overall performance of computer systems with a given set of applications depends on numerous factors and can not be attributed to only the speed of the central processing unit (CPU). Among the most important factors is the connection of the CPU to the computers main memory [6]. In the early days of computers the employed memories were faster or at least comparable in speed to the CPU. Naturally the focus of the evolution of CPUs was to increase their processing speed. This goal was greatly aided and in fact only made possible by continuous downscaling of the dimensions of the devices which they components are built on. This downscaling of the densely packed logic found in CPUs made it possible to attain ever higher clock speeds thereby increasing their maximum performance. The main effect on random access memory (RAM) modules, on the other hand, was to increase their sizes, again by an ever growing level of integration. While speed was also an important concern it quickly lagged behind, as the signal to noise ratios in the highly integrated structures worsen, which leads to increased latencies due to the necessity of appropriate signal handling to insure proper operation.

The reduction of feature size and therefore the increase of operating frequencies is not going to continue without bounds and different strategies have to be used to increase the performance of the processing cores. Nevertheless this trend of increasing clock speeds, especially of CPUs, has already led to the problem that CPUs require data at a faster rate than memories are able to supply. This has resulted in the development of memory hierarchies introducing several levels of caches and instruction pipelines, thereby increasing the overall performance of the systems. The additional complexity induced by these measures makes it more and more important to employ appropriate compilers and techniques to obtain optimal performance [5, 12]. This is even more so as the increase of computing power of future computer hardware is primarily obtained by multiplying the processor cores.

But not only the hardware of computers and the compilers have evolved and now provide a myriad of features, but new methodologies of software development and programming paradigms have also surfaced. Different approaches have focused on the development of high performance libraries for the area of scientific computing such as PETSc [3], CCA [14], or MTL [17].

MODERN PROGRAMMING PARADIGMS

From a software point of view, numerous new paradigms have evolved recently, which allow the synthesis of highly efficient code on modern hardware. It is now the aim to combine the newly provided possibilities in such a way, that an optimal result not only in terms of run-time efficiency, but also maintainability, extendability, portability, and orthogonality of code is attained. While run-time efficiency, maintainability and extendability of code have classically been contradicting goals, with code tuned for high performance often becoming an unreadable maintenance nightmare, the advent of new compilers deploying new optimizers and feature sets has changed this so that high performance code no longer needs to be unreadable [1, 9, 16]. Especially generic programming accomplishes both, a general solution for most of the application scenarios and highly specialized code parts for minor, but also important, scenarios without sacrificing performance [2, 11]. This has already been demonstrated in the field of numerics and yields figures comparable to Fortran [13, 18], the previously undisputed candidate for this kind of calculations.

Based on these techniques we developed a high performance simulation engine based on the SAP tools [15]. With template meta-programming [1], the functional specification can be used very similiar to the original mathematical formulation, as can be seen in this work. Due to this new programming technique and the corresponding evaluation at compile time the calculation associated with the specified equations is highly optimized by the compiler and thereby ensures excellent run-time performance often superior to highly hand-optimized code. In our case C++ was the language of choice, because currently no other language offers sufficient support for all the necessary programming techniques to enable the required level of abstraction.

Our own investigations in the field of compiler optimization and compiler comparison has shown great differences in optimization behavior and run-time performance of modern programming techniques [8].

INTERCONNECT MODEL

Our interconnect simulation tools use the finite element method to discretize the partial differential equations resulting in a system of equations that eventually has to be linearized, and thereafter solved with a preconditioned conjugate gradient algorithm [10]. To give a glimpse on details we consider a typical problem of forming the equation system.

The problem we consider is posed in the following way:

$$\mathcal{L}\Psi := \operatorname{div}(-\varepsilon\,\operatorname{grad}(\Psi)) - \varrho = 0 \qquad \text{in } \Omega \tag{1}$$

$$\Psi - \Psi_D = 0 \qquad \text{on } \partial\Omega\ , \tag{2}$$

where ε denotes the (isotropic) permittivity of the considered domain, which is assumed to be constant in an element of the tessellation. Due to the weak formulation using Galerkin finite elements [19] weighting coefficients for the local element matrices have to be derived for tetrahedra:

$$g_1^e = \varepsilon \frac{K_{11}^2 + K_{21}^2 + K_{31}^2}{\det\mathbf{J}} \tag{3}$$

there, $\mathbf{K}$ is the adjoint matrix of the Jacobian $\mathbf{J}$ which is derived by the affine transformation of the mesh elements to the standard element. Due to operator overloading different mathematical structures such as scalars, vectors, and even matrices can be handled using identical notation. Therewith the transformation into code results in the following code snippet:

```
double g1 = epsilon*(K11*K11 + K21*K21 + K31*K31)/detJ;
```

To assemble the system matrix, a local element matrix has to be assembled: $\mathbf{S}^e$ stands for the local element stiffness matrix which is derived by:

$$\mathbf{S}^e = g_1^e\mathbf{S}_1 + g_2^e\mathbf{S}_2 + g_3^e\mathbf{S}_3 + g_4^e\mathbf{S}_4 + g_5^e\mathbf{S}_5 + g_6^e \tag{4}$$

The corresponding C++ code reads:

```
Se = g1*S1 + g2*S2 + g3*S3 + g4*S4 + g5*S5 + g6*S6;
```

`S1-S6` means the linear form function matrices and `g1-g6` are calculated at the nodes of the tetrahedra in the global coordinate system [4].

PERFORMANCE ANALYSIS

It should be noted that for high precision simulations it is essential to model the simulation domain as exactly as possible. The accuracy and efficiency of a finite element and finite volume simulation strongly depends on the quality of the tessellation of the domain. As a consequence we introduced a comprehensive solid modeling and mesh generation and adaptation approach [7].

Figure 1 presents the example structure under investigation with a coarse mesh for the following performance analysis. In order to obtain sufficiently accurate results, the mesh size typically has to be in the order of 10^4 to some 10^5 nodes.

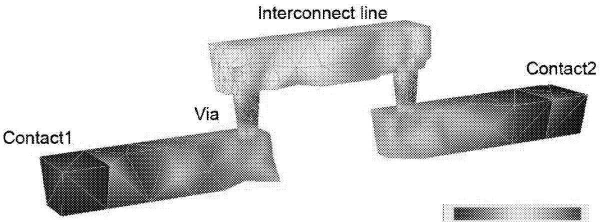


Figure 1: Temperature distribution due to self-heating in a tapered interconnect line with cylindrical vias.

For a rigorous analysis we evaluate three different implementations in C++ and compare them to a hand-optimized Fortran 77 implementation on different computer architectures. The first implementation is based on the GNU GCC `valarray` data-type which is a standardized data-structure representing a mathematical vector. This data-type has shown excellent performance on different computer architectures with recent compilers. Secondly, we utilize the Blitz++ [18] library, which introduced high performance calculation comparable to Fortran 77 directly in C++. Lastly, a naive C++ implementation is used that creates two temporary objects, one for the addition and one for the assignment. As a consequence all elements have to be accessed three times. The tests were performed on four different computer systems:

CPU type	Clock speed	RAM	Compiler	MFLOPS
Pentium 4	2.8 GHz	2 GB	GCC 4.0.2	2310.9
AMD64	2.2 GHz	2 GB	GCC 3.4.4	3543.0
IBM P655	8x1.5 GHz	64 GB	GCC 4.0.2	16361.7
G5	4x2.5 GHz	8 GB	GCC 4.0.0	24434.0

Figures 2-5 compare these different approaches on different hardware architectures. The y-axes is labeled with million operations per second. The vector addition consists of 3 operations, two additions and one assignment.

For vector lengths smaller than 10^4, cache hits reveal the full computation power of the CPU, longer vectors show the limits imposed by memory bandwidth. The poor performance of naive C++ code is indeed remarkable.

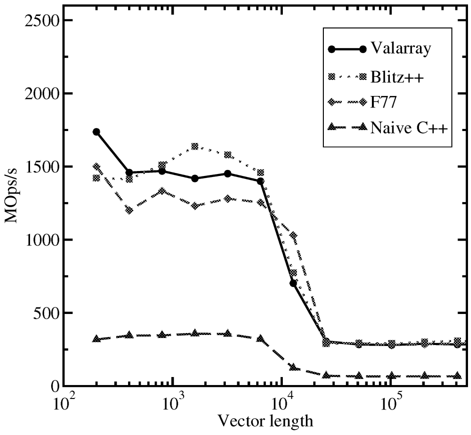


Figure 2: Comparison of different functional specification on the Pentium4.

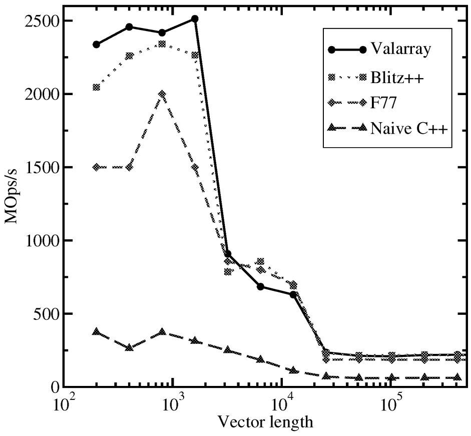


Figure 3: Comparison of different functional specification on the AMD64.

Based on these observations of restrictions due to the limited bandwidth, we illustrate the influence of a problem's size on the overall finite element. We therefore investigate our test structure with different levels of refinement. By resolving a three-dimensional simulation domain, the number of points easily exceeds the critical threshold and thereby leads to severe problems caused by memory bandwidth restrictions (Figure 6).

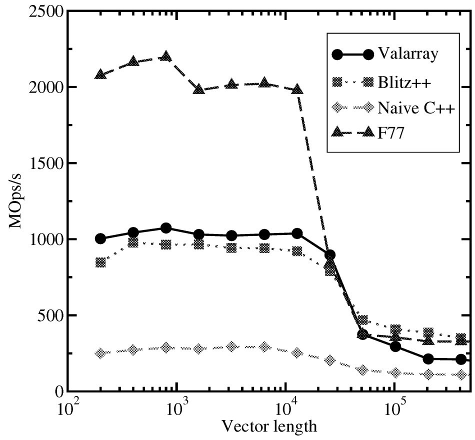


Figure 4: Comparison of different functional specification on the IBM.

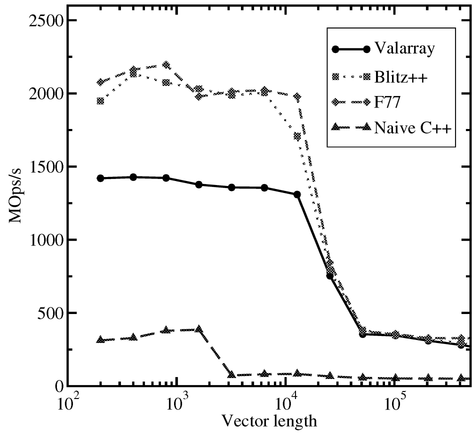


Figure 5: Comparison of different functional specification on the G5.

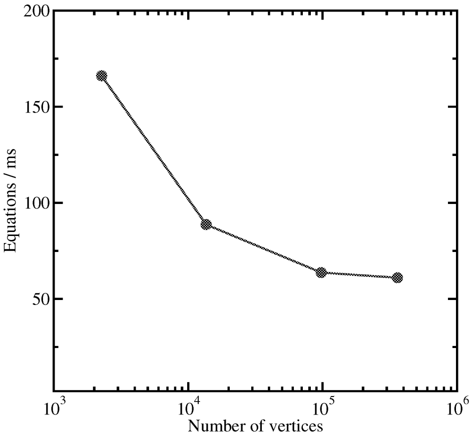


Figure 6: Comparison of the finite element assembly times.

Investigations of parallelization attempts on multiprocessor machines (G5) show that the inner loop of the finite element assembly cannot be parallelized easily. On the one side, the update mechanisms of the element matrices may require access to the same part of memory simultaneously, which could be avoided by a different assembly scheme, e.g. node-based assembly. On the other hand, the inner loops are compiled very efficiently and only bounded by memory bandwidth. Tests with four CPUs have shown, that parallel assembly does not speed up the total assembly process at all. Restrictions resulting from memory bandwidth completly negate any benefit due to parallelization.

CONCLUSION

Although the observed performance issues are presented for the field of interconnect simulation, the main findings are certainly transferable to other areas, such as process and device simulation. Memory bandwidth is the limiting factor as we have seen from our benchmarks.

In summary, highly expressive code in C++ on different platforms and computer architectures does not show any abstraction penalty, where naive C++ code does not perform well. Regarding parallelization, current memory links hardly provide enough bandwidth to accommodate the throughput required to satisfy the computational performance of multiple cores.

REFERENCES

[1] D. Abrahams and A. Gurtovoy. *C++ Template Metaprogramming: Concepts, Tools, and Techniques from Boost and Beyond (C++ in Depth Series).* Addison-Wesley Professional, 2004.

[2] A. Alexandrescu. *Modern C++ Design: Generic Programming and Design Patterns Applied.* Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2001.

[3] S. Balay, K. Buschelman, V. Eijkhout, W. D. Gropp, D. Kaushik, M. G. Knepley, L. Curfman McInnes, B. F. Smith, and H. Zhang. PETSc Web page, 2005.

[4] R. Bauer. *Numerische Berechnung von Kapazitäten in dreidimensionalen Verdrahtungsstrukturen.* PhD thesis, Technische Universität Wien, 1994.

[5] K. Beyls and E. H. D'Hollander. Generating Cache Hints for Improved Program Efficiency. *J. Syst. Archit.*, 51(4):223–250, 2005.

[6] Boost. *Stream - Sustainable Memory Bandwidth in High Performance Computers.* http://www.cs.virginia.edu/stream/.

[7] R. Heinzl and T. Grasser. Generalized Comprehensive Approach for Robust Three-Dimensional Mesh Generation for TCAD. In *Proc. Conf. in Sim. of Semiconductor Processes and Devices*, pages 211–214, Tokio, September 2005.

[8] R. Heinzl, P. Schwaha, M. Spevak, and T. Grasser. Performance Aspects of a DSEL for Scientific Computing with C++. In *Proc. of the POOSC Conf.*, Nantes, France, July 2006.

[9] R. Heinzl, M. Spevak, P. Schwaha, and T. Grasser. A High Performance Generic Scientific Simulation Environment. In *Proc. of the PARA Conf.*, Umea, Sweden, June 2006.

[10] M. Heroux, R. Bartlett, V. H. R. Hoekstra, J. Hu, T. Kolda, R. Lehoucq, K. Long, R. Pawlowski, E. Phipps, A. Salinger, H. Thornquist, R. Tuminaro, J. Willenbring, and A. Williams. An Overview of Trilinos. Technical Report SAND2003-2927, Sandia National Laboratories, 2003.

[11] J. Järvi, J. Willcock, and A. Lumsdaine. Concept-Controlled Polymorphism. In *GPCE '03: Proc. of the 2nd Conf. on Generative Prog. and Comp. Eng.*, pages 228–244, New York, NY, USA, 2003. Springer-Verlag New York, Inc.

[12] D. Lacey, N. Jones, E. Van Wyk, and C.C. Frederiksen. Compiler Optimization Correctness by Temporal Logic. *Higher Order and Symbolic Computation*, 17(3):173–206, 2004.

[13] L. Lee and A. Lumsdaine. Generic Programming for High Performance Scientific Applications. In *JGI '02: Proc. of the 2002 joint ACM-ISCOPE Conf. on Java Grande*, pages 112–121, New York, NY, USA, 2002. ACM Press.

[14] S. Lefantzi, J. Ray, and H. N. Najm. Using the Common Component Architecture to Design High Performance Scientific Simulation Codes. In *IPDPS '03: Proc. of the 17th Symp. on Parallel and Distributed Proc.*, page 52, Washington, DC, USA, 2003. IEEE Computer Society.

[15] Rainer Sabelka and Siegfried Selberherr. A Finite Element Simulator for Three-Dimensional Analysis of Interconnect Structures. *Microelectronics Journal*, 32(2):163–171, 2001.

[16] P. Schwaha, R. Heinzl, M. Spevak, and T. Grasser. Advanced Equation Processing for TCAD. In *Proc. of the PARA Conf.*, Umea, Sweden, June 2006.

[17] J. G. Siek and A. Lumsdaine. The Matrix Template Library: A Unifying Framework for Numerical Linear Algebra. In *ECOOP Workshops*, pages 466–467, 1998.

[18] T. L. Veldhuizen. Arrays in Blitz++. In *Proc. Symp. on Comp. in Obj.-Oriented Parallel Env.*, Lecture Notes in Computer Science. Springer-Verlag, 1998.

[19] O. C. Zienkiewicz and R. L. Taylor. *The Finite Element Method.* McGraw-Hill, Berkshire, England, 1987.

BIOGRAPHIES

RENÉ HEINZL studied electrical engineering at the Technische Universität Wien. He joined the Institute for Microelectronics in November 2003, where he is currently working on his doctoral degree. In April 2005 he achieved first place at the doctoral competition at the EEICT in Brno. His research interests include process simulation, solid modeling, and adaptive mesh generation for TCAD with special emphasis on three-dimensional applications.

PHILIPP SCHWAHA studied electrical engineering at the Technische Universität Wien. He joined the Institute for Microelectronics in June 2004, where he is currently working on his doctoral degree. His research activities include circuit and device simulation, device modeling, and software development.

MICHAEL SPEVAK studied electrical engineering at the Technische Universität Wien. He joined the Institute for Microelectronics in December 2004, where he is currently working on his doctoral degree.

TIBOR GRASSER received the Ph.D. degree in technical sciences, and the "venia docendi" in microelectronics from the Technische Universität Wien in 1999, and 2002, respectively. He is currently employed as an Associate Professor at the Institute for Microelectronics. Since 1997 he has headed the Minimos-NT development group, working on the successor of the highly successful MiniMOS program. In 2003 he was appointed head of the Christian Doppler Laboratory for TCAD in Microelectronics, an industry-funded research group embedded in the Institute for Microelectronics. His current scientific interests include circuit and device simulation and device modeling.

SIEGFRIED SELBERHERR received the Ph.D. degree in technical sciences from the Technische Universität Wien in 1981. Since that time he has been with the Technische Universität Wien as professor. Dr. Selberherr has been holding the "venia docendi" on "Computer-Aided Design" since 1984. As of 1988 he has been chair professor of the Institut für Mikroelektronik. From 1998 to 2005 he served as Dean of the "Fakultät für Elektrotechnik und Informationstechnik" at the Technische Universität Wien. His current topics of interest are modeling and simulation of problems for microelectronics engineering.

DEVELOPING A META-METHODOLOGY SUPPORTING THE APPLICATION OF PARALLEL SIMULATION

László Muka
Elassys Consulting Ltd.
Bég utca 3-5.
H-1026 Budapest, Hungary
e-mail: muka.laszlo@elassys.hu

Gábor Lencse
Department of Telecommunications
Budapest Univ. of Techn. and Econ.
Magyar tudósok körútja 2.
H-1117 Budapest, Hungary
e-mail: lencse@hit.bme.hu

KEYWORDS

meta-methodology, parallel simulation, discrete-event-simulation, organisational process, information and communication technology, Soft Systems Methodology, conceptual model

ABSTRACT

New concepts are described to SSM (Soft Systems Methodology) conceptual models, which are tools for system analysis supporting the application of simulation including decisions about parallel simulation in an organisational environment. A meta-methodology facing with unstructured problems in simulation projects and also supporting parallel simulation is formulated.

INTRODUCTION

Simulation projects initiated to support Information and Communication Technology (ICT) system design and Business Process (BP) design in an organisation usually begin with an unstructured problem situation, where frequently there is on opinion that simulation takes a lot of time and requires significant resources to be assigned with the risk of getting no useful results.

In this paper we outline a meta-methodology addressing these problems: we develop a soft approach to support problem-structuring and effective goal definition to build useful models and also increasing efficiency by precise localization of systems to be modelled and by supporting decisions on the use of *parallel simulation* helping in speeding up the simulation.

In this paper we introduce new concepts to SSM (Soft Systems Methodology, Checkland 1985, 1989) conceptual models then using the new concepts and a traditional six-step process of simulation methodology we outline a *simulation meta-methodology*.

Ideas about N&S (Necessary & Sufficient) conditions and "temporal relations" of conceptual models described by Gregory (Gregory 1993) are used as starting point in our paper.

In (Sierhuis and Selvin 1996, Sierhuis and Clancey 2002) there is a description of *a framework for collaborative modelling and simulation* using SSM and a set of four methods to cover the modelling activities. The main problem with this approach is that there is a *methodological gap* between SSM and methods to deal with simulation.

In our approach this methodological gap is eliminated by the development of modified conceptual models.

DEVELOPING MODIFIED CONCEPTUAL MODELS

The Seven-stage Process of Traditional SSM

Checkland's SSM is an approach to apply systems-thinking to ill-defined problems in human activity systems. It is also described as a system-based problem-solving methodology starting with the unstructured problem situation. By the outcome it is also defined as a learning system, a system for Operational Research or a method for information system analysis and design (Curtis 1989).

Stages of SSM are shown in Figure 1. The process of SSM seems to be linear: it is a sequence of well-defined stages and there is a progression from one stage to the next in the methodology. Working with SSM is an iterative process, since it may be necessary to re-enter an earlier stage for re-execution.

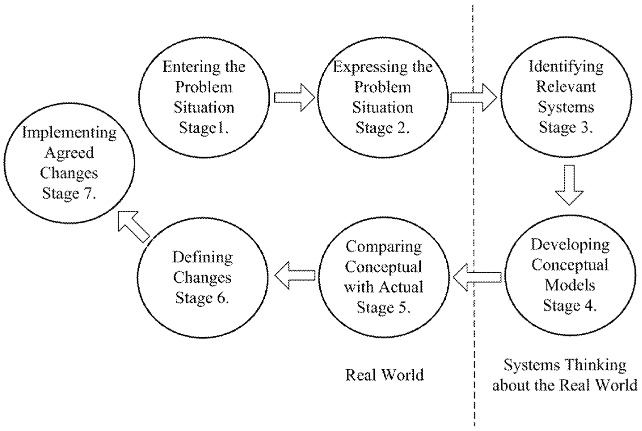

Entering the Problem Situation Stage1.
Expressing the Problem Situation Stage 2.
Identifying Relevant Systems Stage 3.
Developing Conceptual Models Stage 4.
Comparing Conceptual with Actual Stage 5.
Defining Changes Stage 6.
Implementing Agreed Changes Stage 7.
Real World
Systems Thinking about the Real World

Figure 1. The Seven-stage Process of SSM

In Stage 1 and 2 there is a finding out about the unstructured problematical situation that is entering and expressing the problem situation.

In Stage 3 relevant human activity systems are identified and using CATWOE analysis (Checkland 1989) root definitions of selected systems are formulated.

In Stage 4 there is the conceptual model-building of relevant systems from the root definitions provided in Stage 3. Conceptual models are models of the views of what exist and not models of what exist in the real world. In a conceptual model key activities of the system are taken into account. A key activity itself generally represents a subsystem (Curtis 1989) that would carry the activity out, thus a hierarchy of conceptual models can be defined when replacing a first-level conceptual model of a subsystem with its detailed conceptual model.

In Stages 4 - 7 there is a comparison with the real world to define necessary and feasible changes and to define actions to implement changes.

In the following points we harden up the methodology (Jackson and Keys 1984) by introducing new concepts into the conceptual models.

Function elements in Conceptual Models

In this paper we focus on the design of information systems in an organisation therefore we may suppose that a key activity is performed in general by an OP (Organisational Process) function or by an ICT system function. In other words it may be said that any function in the organisation can be performed by some relevant organisational process (P subsystem) with its human resources or by some relevant IT subsystem with its technical resources.

Thus the subsystem elements in our conceptual models can be P-type or IT-type; depending on they represent OP or ICT system function.

In our approach, an important feature of IT elements (according to the traditional approach of SSM) is that any IT element in the model should be connected to a minimum of one P element in order to have its human resource connection. We may look at the conceptual model as a directed graph CM(N;E), where N is the set of nodes containing P-type or IT-type elements, E is the set of directed edges. In order to define the connected feature of IT elements we introduce a logical variable CON to describe that nodes x and y of graph CM are connected:

$$\mathrm{CON}\begin{cases} = 1, \text{if } (x;y) \in \mathrm{E} \text{ or } (y;x) \in \mathrm{E} \\ = 0, \text{otherwise} \end{cases}$$

where $x \in \mathrm{P} \cup \mathrm{IT}$ and $y \in \mathrm{P} \cup \mathrm{IT}$

Now it may be said about IT elements:

$\forall \mathrm{IT}_i$ (i=1;2;…;I) $\exists$j (j=1;2;…;J)
(where I is the number of elements in the set IT, J is the number of elements in the set P)

$\mathrm{CON}(\mathrm{IT}_i;\mathrm{P}_j)=1$

To describe the set of N&S conditions (Gregory 1993) we define three element types F, C and A, It means that there can be PF, PC, PA, ITF, ITC and ITA elements. PF is an element performing basic function in the system; PC is providing conditional function necessary to perform basic function while PA is an agent element ensuring the sufficiency for the basic function to be completed. ITF, ITC and ITA also perform subsequently basic, conditional and agent function, taken into account IT elements' connected feature.

In general, a function is performed if it is assigned to an existing or a new organisational process and the necessary organisational resources (roles and responsibilities) are assigned to the process. It means that using a PA is necessary only in special cases: in the case if the necessary process resources are not assigned in a PF and its PCs elements (for example the necessary resources are assigned in a shared way), or we want to examine the subsystem responsible for the resource assignment.

In the case of an information system design agents can also be IT-type elements, which are software and hardware resources.

Now let us see a short example. Figure 2. shows a conceptual model of a Customer Request Processing System. After receiving the customer request by PF_1 its processing is performed by PF_2, using information obtained by PC_1 from CRM (Customer Relationship Management) database. Customer request is scheduled by PF_3 (service activity assigned to customer request) using schedule information obtained by PC_2 from service department, which is in another system. Answering the request is performed by subsystem PF_4.

PA_1 and PA_2 are agent elements guaranteeing resources for functions in PF_2 and PF_3 to be performed.

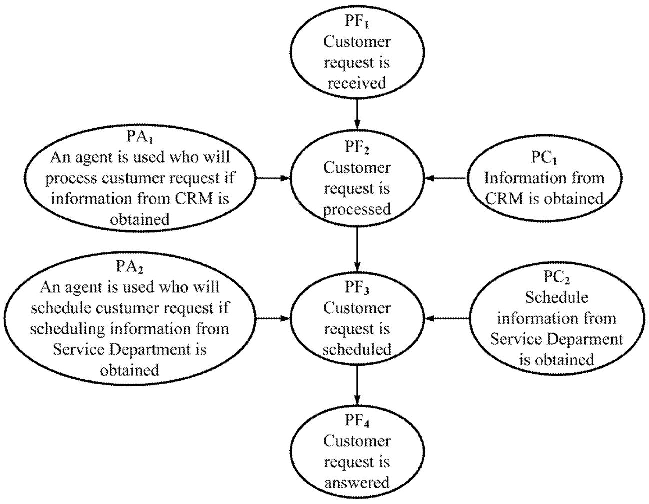

PF₁
Customer request is received
PA₁
An agent is used who will process custumer request if information from CRM is obtained
PF₂
Customer request is processed
PC₁
Information from CRM is obtained
PA₂
An agent is used who will schedule custumer request if scheduling information from Service Department is obtained
PF₃
Customer request is scheduled
PC₂
Schedule information from Service Deparment is obtained
PF₄
Customer request is answered

Figure 2. Conceptual Model with N&S Conditions after Identifying PA, PF and PC Elements

In Figure 3. there can be seen the model of the same system with one agent element PA_1. It was decided not to use PA_2 because in PF_3 and in PC_2 there is a sufficient assignment of resources and we do not want to examine the resource assignment subsystem.
We express N&S conditions in symbols for PF_2 and PF_3:

$\mathrm{PF}_2 \Leftrightarrow (\mathrm{PF}_1 \wedge \mathrm{PA}_1 \wedge \mathrm{ePC}_1)$
$\mathrm{PF}_3 \Leftrightarrow (\mathrm{PF}_2 \wedge \mathrm{ePC}_2)$

In Figure 3. elements PC_1 and PC_2 are expanded (Checkland 1985). PC_1 contains subsystems $ITC_{1.1}$ (the CRM function subsystem) operated by $PC_{1.1}$. In PC_2 there are subsystems $ITC_{2.1}$ and $PC_{2.1}$. where $ITC_{2.1}$ can be an intranet system function and $PC_{2.1}$ a function to provide Service Department's scheduling information obtained using intranet function. The operating subsystem of intranet here is not examined.

The conceptual model CM in Figure 3. can be described as directed graph CM(N;E;TR) where TR is the set of transient edges. Transient edges connect elements in different conceptual models. (A conceptual model we got from an expanded element is also defined to be a different one.) In Figure 3. elements PC_1 and PC_2 are expanded. They contain P-type and IT-type elements in different configurations.

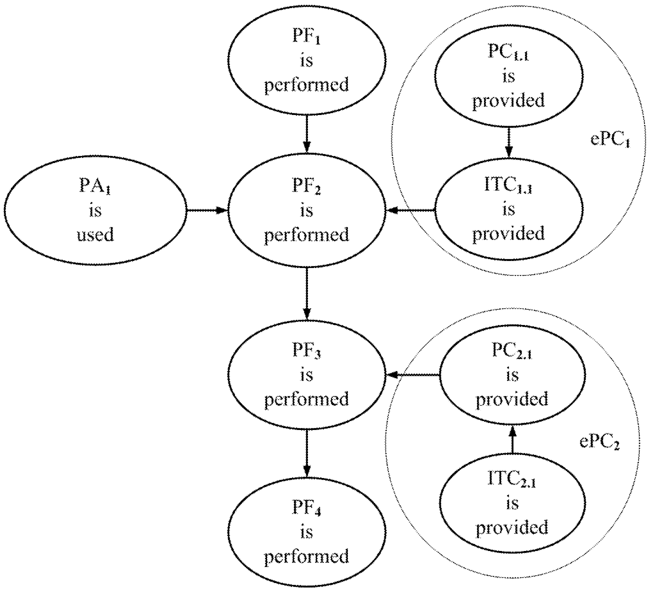


Figure 3. Conceptual Model with Expanded Elements

The expanded elements ePC_1 and ePC_2 are also conceptual models described by directed graphs $CM.ePC_1$ and $CM.ePC_2$ (CM denotes the original conceptual model). Edge $(PC_{1.1};ITC_{1.1})$ in graph $CM.ePC_1$ represents an operator-type connection while edge $(ITC_{2.1};PC_{2.1})$ in graph $CM.ePC_{1.2}$ shows a provider-type connection. An operator P element is responsible for a function of an IT element, while a provider P element is responsible for a function using an IT element. The transient edges $(CM.ePC_1.ITC_{1.1};CM.PF_2)$, $(CM.ePC_2.PC_{2.1};CM.PF_3)$ connect elements of expanded subsystems to elements in conceptual model CM.

Virtual Time and Synchronisation in Conceptual Models

Introducing time into the conceptual models can be done by assigning time label to elements. Giving time label T to an element has the meaning that the event of a 'function is performed' takes place at T.

A conceptual model's **virtual time** is a time sequence assigned to a conceptual model by giving time labels to elements. Time labels $T_{(i)}$ and $T_{(i-1)}$ have the meaning that a function with time label $T_{(i-1)}$ performed earlier than a function with time label $T_{(i)}$. (See in Figure 4.) There is nothing said about the measure $\Delta T= T_{(i)}- T_{(i-1)}$. (To give an estimate of ΔT, simulation method can be applied.)

In Figure 4. we show two conceptual models CM1 and CM2, where CM1 may be the Customer Request Processing System from our previous example and CM2 system performing services (Service Department).

CM1 and CM2 are connected with request and answer connections (RCM2-RCM1, ACM2-ACM1) which may be described as graphs' transient edges

$(CM1.PF_i;CM2.ePC_x.PC_{x(1)})$,

$(CM2.PF_u;CM1.ePC_v.ITC_{v(1)})$

RC (Request from Customer) and AC (Answer to Customer) are entry and exit edges of graph CM1.

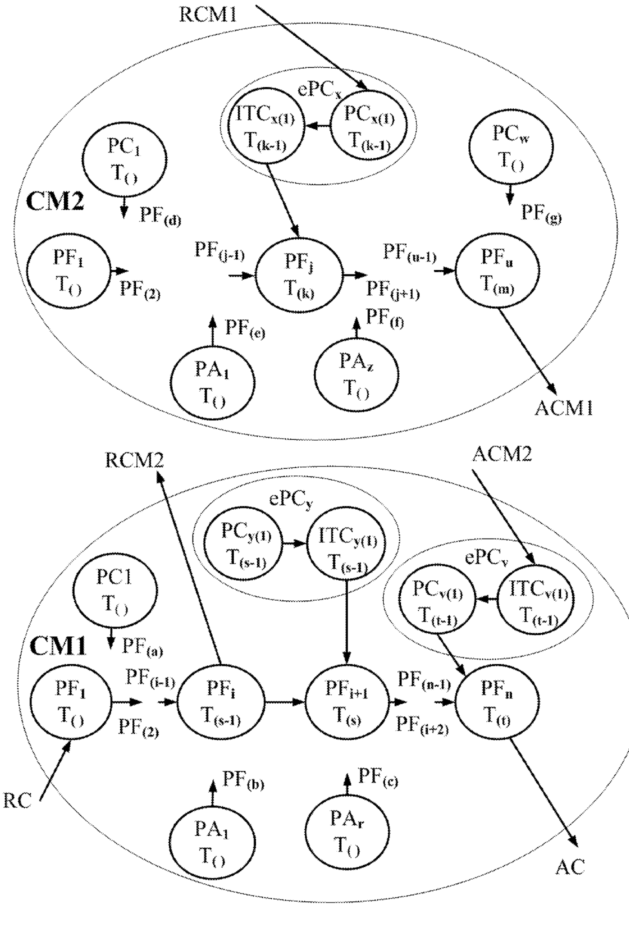


Figure 4. Synchronising Conceptual Models CM1 and CM2 through Conditional Elements

We remark that IT and P elements in expanded subsystems have the same time label. ($T_{( )}$ denotes a time label which is not significant in our analysis.)

Through transient edges and conditional elements (PC_x and PC_v) virtual times of conceptual models are **synchronised**. After synchronisation of CM1 and CM2 we have the next relations:

$T_{(s-1)}=T_{(k-2)}$ and $T_{(s-1)}<T_{(k-1)}$
$T_{(m)}=T_{(t-2)}$ and $T_{(m)}<T_{(t-1)}$

On the bases of synchronisation a decomposition of execution time of functions can be made.

CONSIDERATIONS ABOUT APPLICATION OF PARALLEL SIMULATION

Note: We have not used any constraints on the type of simulation (continuous, discrete, time-driven, event-driven, etc.) therefore our results may be used to the application of Event-driven Discrete-Event Simulation (DES) which is in the focus of our interest.
In the case of information system design after the IT and P function analysis, assigning virtual time and synchronisation of conceptual models, we can have a critical set of elements to be simulated.
The critical set may be an interconnected set of IT and P elements but practically it is a set of at least one IT element connected to one P element. This is the situation to consider parallel simulation.
In the case of one IT and one P subsystem, depending on the focus of the simulation there can be three basic parallel simulation decisions: (1) detailed simulation of both IT and P subsystems; (2) detailed simulation of IT system with simulated P as process environment; (3) detailed simulation of P with simulated IT system as environment. The P and IT parts can act as the two segments of parallel discrete event simulation. They can be executed parallel by two interconnected processors. For all the three situations the use of the Statistical Synchronisation Method (Pongor 1992) can be considered as an inter-processor synchronisation method if there is a relatively slow speed of changes in subsystems' states. In situation (3) the method of TFA (Traffic Flow Analysis) (Lencse 2001) may be appropriate. Methods for the parallel execution of the Combined DES and TFA (Lencse 2004) can be found in (Lencse 2005).
If a subsystem seems to be too complex to be simulated in one model a further partitioning by expansion of the element can be considered.
Expanding a P element we may get a set of P and IT subsystems while an expanded IT may contain only IT subsystems representing a set of sub-functions of the element.
If we have a situation with more IT and P elements grouping or integrating elements may be appropriate.

SIMULATION META-METHODOLOGY SUPPORTING PARALLEL SIMULATION

The six-step process of simulation analysis method

In order to use in formulation of a meta-methodology, in the next point we describe a classic SM (Simulation Methodology).
SM is a six-step process comprising: (SM1) Defining goals (including preliminary design of models); (SM2) Gathering and analysing data; (SM3) Model design and model building (SM4) Performing simulation (with as-is, what-if analysis, model verification and validation); (SM5) Analysing simulation results; (SM6) Supporting implementation.
SM is also an iterative-type methodology which is applicable for both P and IT elements. In point SM1 we explicitly took into account a preliminary model design, which typically takes place only implicitly.
The decision about *parallel simulation* usually is made in step SM3 or SM4.

Outlining the meta-methodology with support for parallel simulation

Now we outline a meta-methodology (MM) applying the new concepts concerning conceptual models introduced in this paper, using the classic SSM together with SSM with modified conceptual models and SM described in previous point.

The phases MM1-MM4 basically follow the progress of SM but in MM2-MM3 there is a soft systems type progress also. In every phase classic SSM is applied if we are facing an unstructured problem and modified conceptual models are applied concerning questions of simulation.
Methodology steps based on our new concepts are listed in MM3.

The phases of meta-methodology are:

MM1. Goal definition

MM2. Identification of a widened set of relevant systems

MM3. Development of conceptual models containing systems to be simulated

- Identify P and IT subsystems, and elements to N&S conditions
- Define time relations in models, synchronise models, make time decomposition
- Define critical P and IT elements to be simulated
- Make decisions on partitioning and grouping of *P elements* for parallel simulation
- Make decisions on partitioning and grouping of *IT elements* for parallel simulation

MM4. Support for implementation

MM1 Phase of defining goals (SM1) has great importance: this is the basis for effectiveness and efficiency. Goals for simulation project should be got from the organisational goals and objectives by the way of goal partitioning and linking to the processes to be simulated. Soft method should be used for learning the situation and for defining requirements for simulation models.
Some fast full simulation cycles may be necessary to make clear the objectives. In this phase methods like TFA may be useful. Preliminary design of simulation models may be produced taking into account the principle of parsimony (Pied 1991).

MM2 In this phase a widened set of relevant systems is identified: (SM2) systems from where data should be get for simulation (to identify and analyse sources of data), systems for which simulation results may be interesting and systems probably to be simulated, that is all systems possibly influenced by the simulation project. During data analysis (SM2) typical and critical data configurations should be defined for the whole interval of simulation, or if possible for a longer time. Identification of typical and critical data configurations should be done for all relevant systems.

MM3 First, conceptual models to be simulated are selected and developed then the new methodological elements are applied.
In the selected models P and IT elements are identified by building up a map for the identified elements. Virtual time system is introduced into conceptual models and after selecting P and IT elements planned to be simulated precise time label values are assigned. Synchronisations of models are made through appropriate conditional elements. On the bases of synchronisation a decomposition of execution time of functions can be made. Now we may have a critical set of P and IT elements to be simulated.
Thinking in parallel simulation we make decisions about further partitioning or grouping elements: in case of too large subsystems we may try to use expansion for partitioning. The use of Statistical Synchronisation Method and TFA can be considered.
After making parallel-sequential decisions the traditional simulation is completed (SM3,4). At these points soft cycles may be necessary to define the what-if scenarios and also for verification and validation of simulation models.

MM4 Analysis of simulation results (SM5) may lead to going back to earlier points for further analysis and simulation or even if the results are satisfactory, the exact understanding may require more soft cycles.
Support for implementation of results (SM6) may consist of making correction plans. In making correction plans, further soft cycles and simulation may be useful.

CONCLUSIONS

We have developed new concepts to SSM to modify conceptual models:
- we have introduced a system of IT and P elements to help common analysis of ICT and BP systems taking into account N&S conditions of performing functions,
- we have defined the virtual time and conceptual model synchronisation concepts for compatibility with simulation methods,
- we have examined how the parallel simulation decision can be supported in conceptual model analysis.

Using the results in developing modified conceptual models we have outlined a meta-methodology dealing with unstructured problems in a simulation project and also supporting the application of parallel simulation.

REFERENCES

Jackson, M.C., Keys, P. 1984. *Towards a System of Systems Methodologies* J. Opl. Res. Soc. Vol. 35, No. 6.

Checkland, P. 1985. *Achieving “Desirable ad Feasible” Change: An Application of Soft Systems Methodology* J. Opl. Res. Soc. Vol. 36, No. 9.

Checkland, P. 1989. *Soft systems methodology* In Rational Analysis for a Problematic World, Edited by J. Rosenhead, John Wiley & Sons Ltd

Curtis, G. 1989. *Business Information Systems* Addison-Wesley, Wokingham, UK.

Gregory, F. 1993. *Cause, Effect, Efficiency and Soft Systems Models* J. Opl. Res. Soc. Vol. 44, No. 4.

Pidd, M. 1991. *Computer simulation methods* In Operations Research in Management, Edited by Littlechild, S., and Shutler. M., Prentice Hall, UK.

Sierhuis, M., Selvin, A.M. 1996. *Towards a Framework for Collaborative Modeling and* Simulation, Workshop on Strategies for Collaborative Modeling & Simulation, CSCW ’96, Boston, MA

Sierhuis, M., Clancey, W.J. 2002. *Modeling and Simulating Work Practice: A Method for Work System De*sign IEEE Intelligent Systems, September/October 2002, Vol. 17, No. 5. pp.32-41.

Elassys Consulting Ltd. 2005 "Iminet Network Expert System" http://www.elassys.hu

Lencse, G. 2001. *Traffic-Flow Analysis for Fast Performance Estimation of Communication Systems* Journal of Computing and Information Technology 9, No. 1, pp. 15-27.

Lencse, G. 2004. *Combination and Interworking of Traffic-Flow Analysis and Event-Driven Discrete Event Simulation* Proceedings of the 2004 European Simulation and Modelling Conference (ESM®'2004) (Paris, France, Oct. 25-27. 2004.) EUROSIS-ETI, 89-93.

Lencse, G. 2005. *Speeding up the Performance Analysis of Communication Systems* Proceedings of the 2005 European Simulation and Modelling Conference (ESM®'2005) (Porto, Portugal, Oct. 24-26.) EUROSIS-ETI, 329-333.

Pongor, Gy. 1992. *Statistical Synchronization: a Different Approach of Parallel Discrete Event Simulation* Proceedings of the 1992 European Simulation Symposium (ESS'92) (Nov. 5-8, 1992, The Blockhaus, Dresden, Germany.) SCS Europe, 125-129.

BIOGRAPHY

GÁBOR LENCSE received his M.Sc. in electrical engineering and computer systems at the Technical University of Budapest in 1994 and his Ph.D. in 2000. The area of his research is (parallel) discrete-event simulation methodology. He is interested in the acceleration of the simulation of communication systems. Since 1997, he works for the Széchenyi István University in Győr. He teaches computer networks and networking protocols. Now, he is an Associate Professor. He does R&D in the field of the simulation of communication systems for the Elassys Consulting Ltd. since 1998.
Dr Lencse works part time at the Budapest University of Technology and Economics (the former Technical University of Budapest). There he teaches digital design and computer architectures.

LÁSZLÓ MUKA graduated in electrical engineering at the Technical University of Lvov in 1976. He got his special engineering degree in digital electronics at the Technical University of Budapest in 1981, and became a university level doctor in architectures of CAD systems in 1987. Dr Muka finished an MBA at Brunel University of London in 1996. Since 1996 he has been working in the area of simulation modelling of telecommunication systems, including human subsystems.
He is a regular invited lecturer in the topics of application of computer simulation for performance analysis of telecommunication systems, at the Széchenyi István University of Győr.

EXPLORATORY MODELING WITH SMALLDEVS

Vladimír Janoušek
Előd Kironský
Faculty of Information Technology
Brno University of Technology
Božetěchova 2, 61266 Brno, Czech Republic
e-mail: {janousek | kironsky}@fit.vutbr.cz

KEYWORDS

DEVS, prototype object, trait, delegation, clonig, reflectivity, Smalltalk, Self, GUI

ABSTRACT

This paper is an introduction to the simulation and modeling framework and tool SmallDEVS. It is a new modeling and simulation framework for Smalltalk. SmallDEVS is different from other tools of its category, because of its openness and reflective features. It supports class-based as well as prototype-based object-oriented model construction. Its meta-object protocol allows the models to be constructed from scratch and inspected and edited during run-time. Interactive modeling and simulation is supported by a graphical user interface which has been highly influenced by the user interface of Self.

INTRODUCTION

This paper introduces a new modeling and simulation tool for the programming language Smalltalk named SmallDEVS. It is an experimental software. SmallDEVS has been designed mainly for experiments with evolving, self-modifying models and with interactive modeling under simulation. The tool is being developed and tested since 2003. As one could already foretaste, SmallDEVS is based on the DEVS (Discrete Event System Specification) formalism. DEVS was introduced in 1976 by Bernard P. Zeigler (University of Arizona). The formalism specifies a system hierarchically. A model can be specified as a coupled model comprising interconnected subsystems, or as an atomic model. Atomic model is a state machine described by its state variable and four functions – external transition δ_{ext}, output function λ, internal transition δ_{int}, and time advance ta. The theory behind DEVS comprises also abstract simulators for atomic and coupled models.

DEVS makes systems modeling and simulation clear and easily understandable whereas it keeps a relatively simple structure. There are many variants of the DEVS formalism. This text will consider only the classical version of DEVS.

Since the invention of the DEVS formalism, new implementations for various programming languages are coming up. Most of these programming languages are object oriented like C++ or Java. SmallDEVS package is a DEVS implementation for Smalltalk. While SmallDEVS allows us to implement a model as a class in a traditional fashion, we prefer the use of prototype objects to create models because of a higher flexibility of this approach. The creation of the models and the experimentation with them is supported by a graphical user interface which is highly influenced by the user interface of Self, a prototype-based object oriented language and system (Ungar and Smith 1989). More concretness and more interactivity in the construction of models of discrete-event systems—these are the main ideas behind SmallDEVS development. It is our believe that the implementation of these ideas can significantly contribute to the quality of the “understanding by modeling”.

Our motivation to design and implement a new simulation and modeling tool is discussed in the next sections. We will explain, why did we choose the Smalltalk programming language and what are the innovative features of SmallDEVS. We assume, that the reader is already familiar with the details of the DEVS formalism (Zeigler at al. 2000) and therefore we will skip it.

CLASS-BASED DEVS IMPLEMENTATION

The majority of DEVS modeling and simulation tools is implemented in C++ (Zeigler at al. 1996) or Java (Zeigler at al. 1997). An implementation of a DEVS model in a class-based object-orented languages obviously leads to the modeling by subclassing the existing models. The subclasses can define new instance variables for the representation of state; redefine the methods corresponding to the four main functions (internal transition, external transition, output function and time advance function) of atomic models; and specify a component list and a coupling relation for the coupled models. An initialization method is responsible for creating input and output ports.

SmallDEVS supports the class-based approach to the modeling in a way that is very similar to that of Python DEVS (Bolduc and Vangheluwe 2002). Both of these frameworks are very close because they are implemented in dynamically typed languages. Nevertheless, this paper deals with a more flexible approach—a prototype-based model construction, which is explained in the next section.

PROTOTYPE-BASED DEVS IMPLEMENTATION

The class-based modeling brings some complications into the play when we deal with evolving and self-modifying models. Especially the DEVS implementations which are built using statically compiled languages such as Java and C++ are very limited in their flexibility because all the code which could be possibly needed has to be known at the compile time. Dynamic modifications to a model during a simulation are limited to the structural changes only. Every time we want to modify the behavior of an atomic model, we must recompile the code of the coresponding class and restart the simulation.

The traditional approach to the dynamic model implementation relies on a well-designed set of fine-grained atomic models. The dynamics is then expressed by the structural changes of the coupled models. Dynamic languages are more flexible. They allow also the atomic models to be dynamicaly changed during runtime. Nevertheless, even the dynamic class-based object-oriented languages do not offer enough flexibility. For example, if we have several instances of the same model and we want to change only one of them in a specific way, most likely we have to define a separate class for it. It is not an essential problem, but it is a complication which can be easily eliminated by switching to the prototype-based (i.e. classless) approach.

SmallDEVS is implemented in Squeak Smalltalk (Ingalls et al. 1997) using an extension, that allows to modify the structure and behavior of the individual instances. This extension is installed with the package Prototypes. This package makes possible to create prototype objects. A prototype object can be created as an instance of the class PrototypeObject, or as a clone of another prototype object:

$aPrototypeObject := PrototypeObject\ new.$
$anotherPrototypeObject := aPrototypeObject\ clone.$

The class PrototypeObject defines a protocol that allows us to edit slots and methods for any particular prototype object without a need to define a new class for it:

$aPrototypeObject\ addSlots : \{$
$'name1' - > anObject.$
$'name2' - > anotherObject\}.$

$aPrototypeObject\ addMethod :$
$'messageSelector\ ...(method\ body)...'.$

Values of the slots can be accessed by sending the appropriate messages to the objects, e.g. $self\ slotName$, or $self\ slotName : aValue$.

Shared behavior can be specified by means of *traits*. Traits are prototype objects which contain methods which are intended to be shared (dynamically inherited) by other objects (models). Other objects can delegate messages to them (it is also refered to as the dynamic inheritance or the instance-based inheritance). The delegates (traits) can be specified by the delegation slots:

$aPrototypeObject\ addDelegates : \{$
$'name1' - > aTrait.$
$'name2' - > anotherTrait\}.$

Note that the traits can also delegate parts of their behaviour to some other traits. This way, the traits can play the role of classes and the delegation can play the role of inheritance. Also note that a multiple delegation is possible, as well as a runtime changes of the delegates. We can see that no feature of class-based object-orientation has been lost. What is more important, the prototype-based object-orientation offers more flexibility which is needed for interactive modeling and model refactoring. The prototype objects can behave completely differently depending on their slots and methods which can be incremetally edited at run-time (we can add and also remove slots, methods, and delegates). This feature opens a huge box of possibilities. The atomic models can be created in a very simple way from prototypes by adding slots, delegates and methods and then they can be modified dynamically. Such a degree of flexibility is needed to support

- reflective and evolving systems modeling – as an example, we can mention anticipatory systems (Rosen 1985), which perform nested simulations of themselves, possibly with some modifications, in order to support their decisions about their next actions and self-improvements;
- interactive and incremental construction of the models under simulation – we call it *exploratory modeling*, similarly to the notion of exploratory programming which represent the way of programming in Smalltalk which is based on wast exploration of the actual state of a running program, together with the program modifications during runtime.

Reflectivity is an essential feature of SmallDEVS – we can not only build a model incrementally, but we can also inspect what has been actually built (what is really needed if we allow models to evolve automatically) and in which state the simulation is. Anything we can do interactively, the models can do themselves as well. This leads to an interesting area of reflective systems modeling and simulation.
When we used class-based approach, we needed classes to be created and discarded during the model evolution. Smalltalk can be used this way (classes can be created even on the fly, without registering in the smalltalk class repository), but the prototype-based approach is much simpler (we don't have to deal with classes if we don't need them) and more flexible (thanks to the dynamic inheritance and individual objects modifications).

SmallDEVS allows an atomic DEVS to be created by executing the following expressions:

$model := AtomicDEVSPrototype\ new.$

$model\ addSlots : \{'name' - > value....\}.$
$model\ addInputPorts : \{'name1'.\ 'name2'....\}.$
$model\ addOutputPorts : \{'name1'.\ 'name2'....\}.$
$model\ addDelegates : \{'name' - > aTrait\}.$
$model\ intTransition :\ '...(method\ body)...'.$
$model\ extTransition :\ '...(method\ body)...'.$
$model\ outputFnc :\ '...(method\ body)...'.$
$model\ timeAdvance :\ '...(method\ body)...'.$

A coupled DEVS can be created by executing the following expressions in SmallDEVS:

$model := CoupledDEVSPrototype\ new.$
$model\ addInputPorts : \{name1.name2....\}.$
$model\ addOutputPorts : \{name1.name2....\}.$
$model\ addComponents : \{name - > aComponent....\}.$
$model\ addCouplings : \{$
$\#(component1\ port1) - >\ \#(component2\ port2).$
$\#(component3\ port3) - >\ \#(component4\ port4)....\}.$

OPERATING SYSTEM

SmallDEVS has been designed specifically for the experimenting with several interesting techniques such as multisimulations, reflective systems simulation studies, interactive modeling and simulation, and model based design. The following requirements were set on SmallDEVS:

- manipulation with models (create, delete, inspect, edit),
- manipulation with simulations (run, stop, resume, inspect, cloning, saving and restoring simulation state, nested simulations),
- it should be no difference between manipulating models in a running simulation or separately,
- interactivity and visualization.

As to the manipulation with models, the abstract examples in the previous section showed how to create models incrementally. Beside that we are able to aquire any detail about the model - slot names and content, method sources, ports, delegates, components, couplings and we can also remove them and/or edit them. This makes full inspecting and editing of our model possible. These changes can be made either interactively or programatically. The same operations are also available during a running simulation. SmallDEVS makes sure, that the editing operations are executed safely between the simulations steps. One can also synchronize the simulation with real-time to support interactive and HIL simulations.

Simulations can be started and controled in a following way:

$aSimulation := aModel\ getSimulatorRT.$
$aSimulation\ stopTime :\ Float\ infinity.$
$aSimulation\ RTFactor :\ 1.$
$aSimulation\ start.$
$aSimulation\ stop.$

The simulation runs on the background and it is possible to start and control more simulations simultaneously. The models, their parts, as well as the simulations, can be cloned:

$aModel2 := aModel1\ copy.$
$aSimulation2 := aSimualtion2\ copy.$

A copy of a simulation creates a copy of the complete state of the simulation. Note that a copy of a model can be made at any time during the simulation, of course. What is important, any copy of the model made during the simulation can be used as an initial state for another simulation, and/or saved as a text for possible editing of the code by hand.

An important responsibility of SmallDEVS operating system is persistency. On the basic level, any object pointed to by some Smalltalk Workspace variable is persistent (and can be stored as a part of the Smalltalk object memory). Nevertheless, for serious work it is not sufficient and SmallDEVS offers a better solution. The models, as well as the simulations (either running, or ready-to-run) can be stored and organized in a structure named MyRepository (the name is relatively general because it is intended not only for models and simulations). MyRepository represents a hierarchy of folders and objects. Objects which are considered to become patterns for cloning can be put among other well-known objects, into the objects tree (and available by a pathname in MyRepository) as prototypes. This tree is unique in the system and is rooted in Smalltalk as a global variable. Generally, MyRepository can hold any object, that understands a protocol allowing for hierarchical composition of objects and folders. The inspiration came from filesystems of traditional operating systems. The main difference from files in such systems is the fact that objects are live entities residing in Smalltalk object memory, while files are nothing but named strings of bytes lying passively on some external media. Although the SmallDEVS objects can be ”externalized” using XML or as a storeString (a Smalltalk code which, when executed, recreates an exact copy of the original object), their primary form is the live form in the object memory of Smalltalk. Thus, they can be stored and restored at once—in a form of the so-called image—as it is in Smalltalk obvious. Objects (simulations, models, as well as their components) can be accessed in the following way:

$MyRepository\ at :\ '/Sims/TestSim'\ put :\ system.$
$aComp := MyRepository\ at :\ '/Sims/TestSim'.$

The overall structure of the SmallDEVS system is depicted in Figure 1. The lowest level of the SmallDEVS

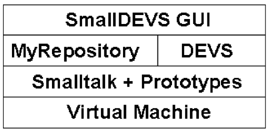


Figure 1: SmallDEVS system

system is the Squeak Smalltalk Virtual Machine. It is responsible for the interpretation of Smalltalk Image (the place where all the objects reside). VM is implemented in a small portion of the C language, being completely portable to almost all known platforms. Smalltalk image with Prototypes package represent another level of the system. The core parts of SmallDEVS are MyRepository and the DEVS simulator. The core contains all the classes that implement the real time simulator and the wrapper classes, that define the reflective interface to the inner prototype objects (traits, atomic models, etc.). The topmost level represent the SmallDEVS GUI which is described in the next sections.

VISUAL TOOLS FOR EXPLORATORY MODELING

The feeling of concretness of the prototype-based approach can be significantly amplified by an approporiate graphical user interface. The SmallDEVS GUI has been higly influenced by the GUI of Self, a prototype-based object oriented language. Self's GUI with direct manipulation of objects significantly amplifies the concretness which is inherent in prototype-based programming. Self's objects can be inspected and modified by the so-called outliners. In fact, the outliner is a merge of inspector and browser, which follows the fact that a prototype object is a standalone object – it has its own data and methods. Self's GUI is able to visualize inter-object relations and modify them in a drag-and-drop manner. Object refactoring (moving slots between objects) is also supported in the same, intuitive and concrete way.
Another inspiration for the SmallDEVS GUI came from the file managers known from the traditional operating systems - they allow creating, copying (by cut/copy/paste actions), renaming and opening files which are organized in folders. In our case, we use this approach to the objects organized in MyRepository.

SmallDEVS allows the models to be created by the GUI or without it. They can be generated by a program. The user interface can visualize and manipulate a model despite the way how the model has been created. The visualization if completely transparent in SmallDEVS. The model components are primary entities, while the user interface is secondary. GUI can be opened on any component of a model at any time thanks to the reflective interface of the models. Each component under investigation has its own, independent GUI instance. The main components of the SmallDEVS graphical user interface are described in the following sections.

MYREPOSITORY BROWSER

MyRepository Browser provides an access to the context menus of the objects. MyRepository is used mainly as a container for models and simulations, but it can contain basically any object (for example documents, pictures, binary data, etc.). Figure 2 shows the window of a MyRepository Browser. You can see the hierarchical tree of objects as well as several simulations in the 'Simulation' directory (subtree).

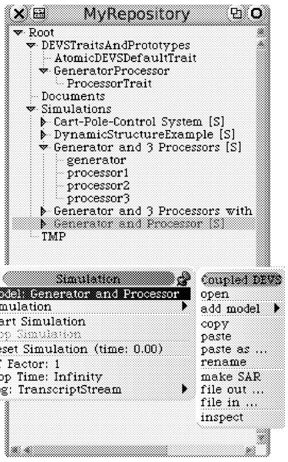


Figure 2: MyRepository browser

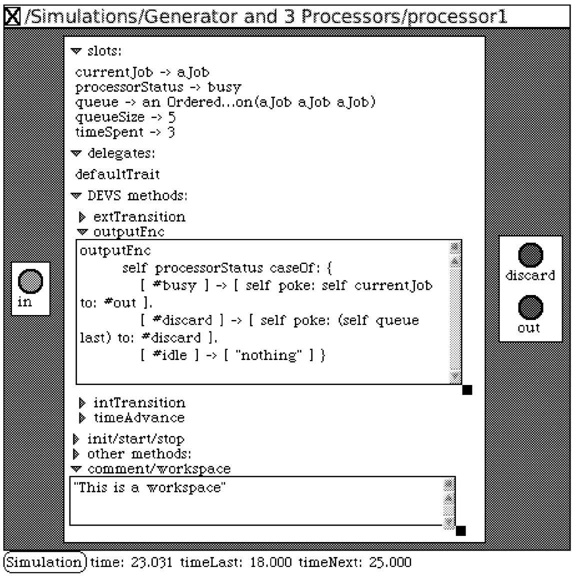
At the bottom of the figure is an opened context menu of the simulation named 'Generator and Processor [S]' that was stopped at the time of the screenshot. From the simulation context menu, the coupled DEVS context menu is poped up, where you can see the operations available on the coupled model.

ATOMIC MODEL INSPECTOR

One of the two main SmallDEVS tools is the atomic model inspector and editor. It is a tool that makes the implementation of atomic models more user friendly, but one still has to write the implementation of the model's behaviour. The design of this editor is heavily inspired by the Self language and its outliner. It allows us to define slots for the model, assign vaues to them, define the four basic methods of a DEVS model and to add other methods when they are needed. A special initialization method is prepared for the user, that is executed at the moment, when the simulation is restarted (or at the first run). This method ensures, that every model in a simulation can be returned to initial state at once or separately when needed. A so called "Workspace" is also part of the tool, where arbitrary expression can be evaluated interactively in the context of the inspected object. This way, the values of the slots can be changed (among other things like the execution of scripts, etc.). A screenshot from a editor of a simple atomic model is in Figure. 3. Notice the expandable editors of methods. The header contains the full path within the MyRepository hierarchy. It is possible to add or remove input ports on the left side and output ports on the right. Also, the bottom status bar provides information about the simulation and the simulation control is accessible from here, too.

COUPLED MODEL INSPECTOR

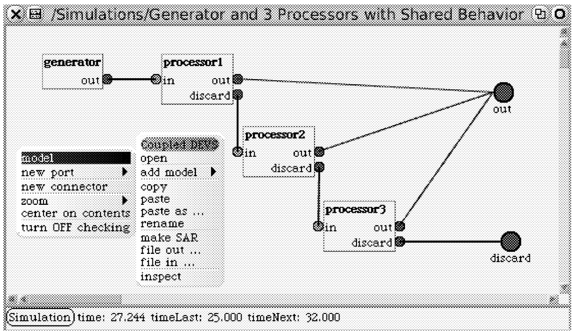
The coupled models inspector is a tool to build, inspect and edit coupled models. The connections editing, together with copying, cutting, pasting, and renaming of the models are supported. The viewable area can be zoomed in or out. Ports and new atomic/coupled models can be added and removed. Also there is an option to choose a model from existing mod-



```
/Simulations/Generator and 3 Processors/processor1
▼ slots:
 currentJob -> aJob
 processorStatus -> busy
 queue -> an Ordered...on(aJob aJob aJob)
 queueSize -> 5
 timeSpent -> 3
▼ delegates:
 defaultTrait
▼ DEVS methods:
  ▶ extTransition
  ▼ outputFnc
outputFnc
      self processorStatus caseOf: {
         [ #busy ] -> [ self poke: self currentJob
to: #out ].
         [ #discard ] -> [ self poke: (self queue
last) to: #discard ].
         [ #idle ] -> [ "nothing" ] }
  ▶ intTransition
  ▶ timeAdvance
▶ init/start/stop
▶ other methods:
▼ comment/workspace
"This is a workspace"
Simulation time: 23.031 timeLast: 18.000 timeNext: 25.000
```

Figure 3: Inspector of atomic models

els in the hierarchy of models and copy that particular model. Like in the editor of atomic models, here is also a status bar with the same function. Figure 4 shows an editor over a simple coupled model.



```
Simulation time: 27.244 timeLast: 25.000 timeNext: 32.000
```

Figure 4: Inspector of coupled models

SUMMARY

The paper showed why and how the prototype-based object orientation can help us to build a tool which can support structurally dynamic and evolving DEVS models and exploratory modeling. SmallDEVS is a highly interactive tool for modeling and simulation. Its real power is in rapid prototyping of DEVS models. It supports model modifying during simulation (interactively as well as programmatically). Generally, SmallDEVS is designed to allow vast experimentations with a model without having to recompile it and start over the simulation each time the model changes. Persistency of models and simulations is also supported, as well as interconnecting models with real components (hardware in the loop). An interesting topic of the future research and development is a meta-language, that could describe DEVS models independently on the underlying software and hardware architecture. This would allow us to develop models and debug them in SmallDEVS and then simulate them on a performance optimized simulator in C++ or on a distributed simulation engine. For intelligent systems simulation, we plan to develop a library of soft-computing components. We also plan to allow the atomic models to be specified by other formalisms such as Petri nets and state charts. Other fields of interests are some applications of the model continuity concept in the intelligent systems development. The current version of SmallDEVS is available on its web site *http://www.fit.vutbr.cz/~janousek/smalldevs*.

ACKNOWLEDGEMENT

This work has been supported by the Grant Agency of Czech Republic grant No. 102/04/0780, ”Automated Methods and Tools Supporting Development of Reliable Concurrent and Distributed Systems”.

REFERENCES

Bolduc, J. S. and H. Vangheluwe. 2002. ”A modeling and simulation package for classic hierarchical DEVS”. Internal document for the MSDL, School of Computer Science, McGill University

Ingalls, D.; Kaehler, T.; Maloney, J.; Wallace, S.; Kay, A. 1997. ”Back to the future. The story of Squeak, a practical Smalltalk written in itself.”. *OOPSLA ‘97 Conference Proceedings*, 318-326.

Ungar, D. and Smith, R. 1989. ”SELF: The Power of Simplicity”. *OOPSLA ‘87 Conference Proceedings*, 227-241.

Zeigler, B. P.; Y. Moon; D. Kim; J. G. Kim. 1996. ”DEVS/C++ A High Performance Modelling and Simulation Environment.”. *29th Annual Hawaii International Conference on System Sciences*, IEEE Computer Society, 350-359.

Zeigler B. P. 1997. ”DEVS-JAVA User’s Guide”. Technical Report, AI & Simulation Lab, Department of Electrical and Computer Engineering, University of Arizona, Tucson.

Zeigler, B. P.; H. Praehofer; T. G. Kim. 2000. *Theory of Modeling and Simulation Second Edition*. Academic Press. ISBN 0-12-778455-1.

BIOGRAPHY

VLADIMÍR JANOUŠEK received the Ph.D. degree from the Faculty of Information Technology, Brno University of Technology in 1999. He is an assistant professor in the Department of Intelligent Systems at the Faculty of Information Technology, Brno University of Technology. His research focuses on simulation-driven development, pure object orientation and reflective architectures.

ELŐD KIRONSKÝ received the M.S. degree from the Faculty of Information Technology, Brno University of Technology in 2005. He is a Ph.D. student in the Department of Intelligent Systems at the Faculty of Information Technology, Brno University of Technology. His research focuses on modeling and simulation tools, robotics and exploratory modeling.

DISTME: A GENERIC TOOLKIT FOR STOCHASTIC SIMULATION DISTRIBUTION

Romain Reuillon, David R.C. Hill
ISIMA/LIMOS UMR CNRS 6158,
Blaise Pascal University

BP 10125 Campus des Cézeaux
63173 AUBIERE CEDEX FRANCE

E-mail: reuillon@isima.fr

KEYWORDS
Distributed Stochastic Simulation, Multiple Replications In Parallel, Parallel Random Number Generator

ABSTRACT

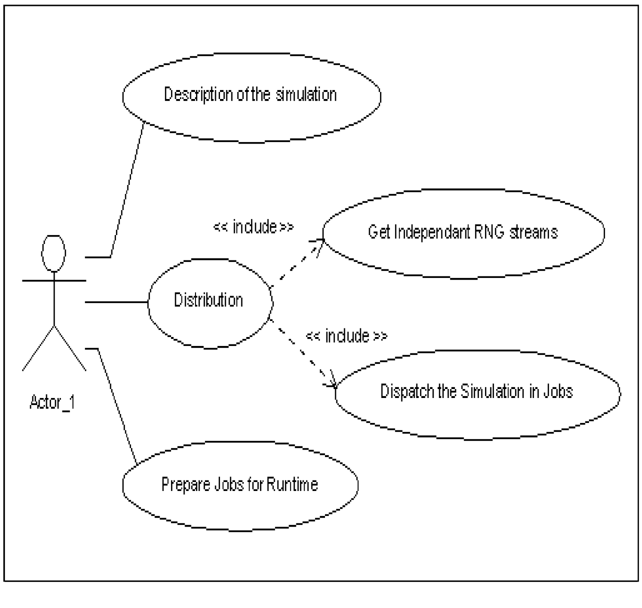
Stochastic simulations are considered as naturally parallel, because many replications of the same experiment may be distributed on multiple execution units to reduce the global simulation time. However, one need to take care of the underling random number streams and ensure the lack of intra and inter stream correlations that can lead to erroneous results. Based on generic random number generator statuses formats and automatic tools to distributed stochastic simulations, DistMe is a java toolkit fully usable to speed up Monte Carlo simulations using any parallel machine based on the bag of work paradigm. A nuclear physic application and an environmental simulation software have been parallelized with this toolkit and results are very convincing.

INTRODUCTION

In agreement with some authors (Mascagni and Srinivasan 2004), we think that Monte Carlo Simulations (MCS) can be considered as naturally parallel. It is widely assumed that with N processors executing N replicates of a Monte Carlo calculation, the pooled result will achieve a variance N times smaller than a single instance of calculation in the same lapse of time (Mascagni et al. 2000). Automatic parallelization is a tough research domain particularly when dealing with fine grain scheduling (Darte et al. 2000). In this paper we present a software toolkit: DistMe which uses this concept to speedup in a generic manner any stochastic simulation at a coarse grain level. We present first the user requirements for simulation distribution software, along with the current state of the art in techniques for stochastic simulations distribution with their targeted execution environments. Then, we expose the interest to establish a standard format for random number generation algorithm statuses and the architecture of our distribution application called DistMe. At the end of the article we present a real simulation case study distributed on several execution environments using DistMe.

REQUIREMENTS FOR DISTME

We have design the DistMe toolkit considering stochastic simulations as black box software pieces using random number streams. The use cases presented in figure 1 illustrate the distribution process. The user can and has to describe the simulation application to distribute. The creation of jobs ready for runtime is then delegated to the DistMe code. The job generation part guarantee a rigorous distribution by respecting the constraints linked to this kind of parallelization.


Description of the simulation
<< include >>
Get Independant RNG streams
Distribution
<< include >>
Dispatch the Simulation in Jobs
Actor_1
Prepare Jobs for Runtime

Figures 1: UML (Unified Modeling Language) use case diagram for DistMe

STOCHASTIC SIMULATION PARALLELIZATION

Parallel computing is still an important research topic, particularly when dealing with stochastic simulation applications. Several approaches are available to speed up simulations with a distributed environment. For deterministic simulations, the High Level Architecture (Rycerz 2005) is a very interesting approach; optimistic parallel discrete event simulation can also be achieved with parallel DEVS (Zeigler et al. 2000) or with direct approaches (Iskra et al. 2005). For quantitative Monte Carlo simulations the MRIP or “Multiple Replication In Parallel” parallelization approach is in use since the beginning of the nineties (Sunderam and Rego 1991) (Rego and Sunderam 1992) and (Pawlikowski and Yau 1992). Still recently used (Pawlikowski 2003), it allows a maximum speed up if many replications of the same experiment have to be made in order to obtain a good approximation of the result. However, one needs to take care in parallelizing the underling pseudo-random number generator (RNG) to avoid correlations within and between the random numbers streams generated in each processor (Mascagni and Srinivasan 2004). Different parallel generation techniques of pseudo random numbers can be found in this survey (Traore and Hill 2001).

In the "sequential approach", a central RNG generator provides numbers for all simulation jobs. This approach is the natural one but doesn't fulfill the requirements for a good parallel RNG (Coddington 1996). The "sequence splitting" or "blocking" consists in splitting the RNG cycle into non-overlapping contiguous sections (Coddington 1996). The "leap frog" technique distributes the sequences to the processor like a deck of cards to card players. This last technique requires a generator that allows cycle division (Srinivasan et al. 1999). The "independent sequences" technique is available for some generators that produce different cycles of numbers depending on the initial seed (Coddington and Newell 2004). This last technique is close to the "parameterization" technique used by Mascagni's team for SPRNG (Mascagni et al. 2000) and by (Matsumoto and Nishimura 2000). It can be might be used with some RNG like the Mersenne Twister (Matsumoto and Nishimura 1997), its principle is to generate algorithm parameters leading to the generation of highly independent random number streams.

Within the current state of the art, we are not able to provide a theoretical proof of independence between pseudo-random number streams. However, the various approaches can be tested empirically, implying heavy computation that can be achieved once for many applications under a precise experimental framework.

TARGETED ENVIRONMENTS

The DistMe is aimed to be adaptable for potentially all kind of execution environment, and especially the ones that fit the bags of work paradigm. In its current version, DisteMe generates basic scripts (using ssh for instance), bags of work for the European Grid using JDL descriptors (the European grid Job Description Language), and also scripts for "OpenPBS" (the open Portable Batch System).

GENERIC RNG STATUSES

Random number generators used for simulation are deterministic algorithms. The status of their parameters can be saved, and restored to initialize the generator at a specific position in the random number cycle. Our work relies on the RNGs proposed by the CLHEP library (Lönnblad 1994). In this library the polymorphic method "saveStatus" of the "HepRandomEngine" class (the superclass for all generators) allows writing the current state of any pseudo-random number generation algorithm implemented in CLHEP in a file called a status file. Those status files are written in a specific format, they can be used to initialize only CLHEP objects. We know from our previous work (Maigne et al. 2004) that the status generation phase might be time consuming, notably for "sequence splinting" techniques. Likewise to ensure the good quality of the parallel random numbers streams, statistical tests have been run. In addition to the famous DieHard tests (from Marsaglia), the recent set of test named testU01 and proposed by L'Ecuyer's team have been achieved (L'Ecuyer and Simard. 2003). Such testing is very computer intensive. To preserve the work done during the PRNG parallelization phases, we have designed a generic format for the most common random numbers generator algorithms. The resulting statuses are now saved in XML files containing the names and values of RNG algorithm parameters. The XML file format for the Mersenne Twister 19937 algorithm is presented in Figure 2. This file contains enough information to be converted into statuses for any random numbers generation library implementing the 32 bits version of the Mersenne Twister 19937 algorithm.

The root XML tag is "status". Then meta-parameters describe the distance in terms of number of drawings between this status and an arbitrary "zero point" status in the sequence and they also provide identification for sub-sequences if the algorithm supports several subsequences (this parameter is for future use). In the "data" XML tag the type of the algorithm is codified and the values and names of its parameter are exposed.

```
<Status>
  <distance>9.456E13</distance>
  <subSerieID>0</subSerieID>
  <data class="MT19937v32">
    <seed>9876</seed>
    <mti>
      <value index="0">3818875949</value>
      <value index="1">4264411326</value>
      <value index="2">959639042</value>

      ...

      <value index="622">3558110699</value>
      <value index="623">2797295466</value>
    </mti>
    <mt>416</mt>
  </data>
</Status>
```

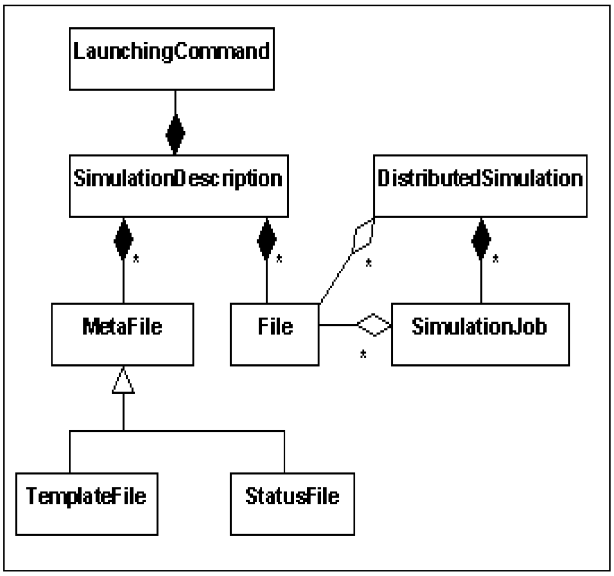
Figures 2: Mersenne Twister XML Status Example

DISTME DESIGN

. The DistMe toolbox contains a status database and is able to create jobs for various distributed environments independently from the random number generation library, thanks to the XML format for RNG statuses. The prototype of this toolbox is available on SourceForge and is in development phase. The first working version with open sources and tutorials can be found on the Internet[1]. Since the aim of the DistMe toolbox is to ease the parallelization of stochastic simulations in distributed environments, the early version of this software parallelizes MCS using the MRIP approach. The challenge is to provide a design at an abstraction level sufficiently high to work with any kind of stochastic simulations and distributed environments.

The Simulation Package

Figure 3, presents an excerpt of the DistMe architecture with a UML class diagram. An instance of the "SimulationDescription" class allows the user to describe his original simulation.

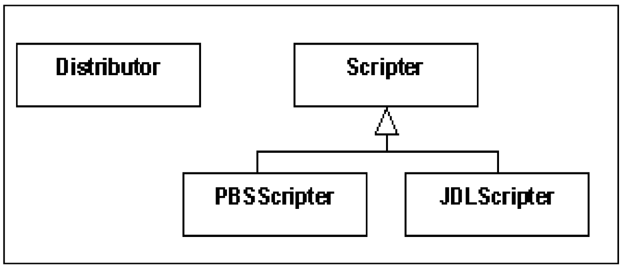


Figures 3: UML class diagram the Simulation package in DistMe

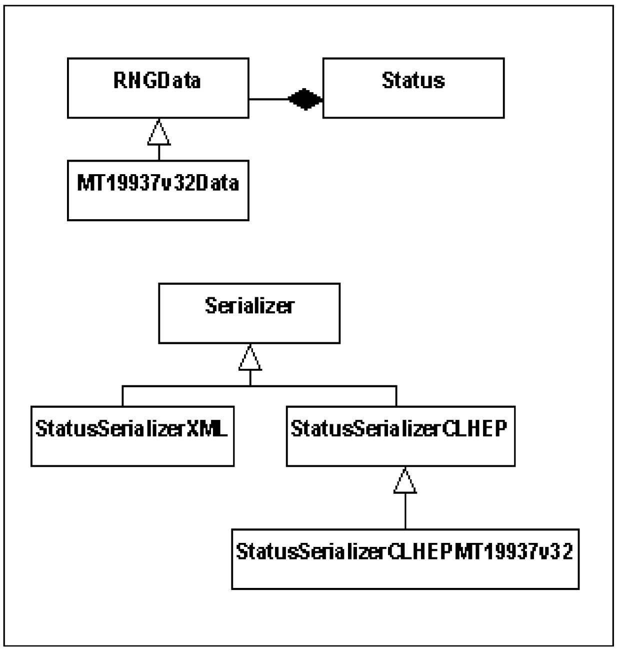
This description has been designed to fit the bag of work paradigm. The user describes every files needed to run the simulation and the launching command. Some input files for the simulation already exist, they are described by the "File" class. Some files are created during the distribution process, they correspond to files with specific data for one simulation job, and they are described by classes deriving from "MetaFile". Such files don't exist before the distribution process and hence cannot have a name at that time. The future name of such files can be used as parameter. For example, statuses files in a specific format for a target library of random numbers generators are created during the distribution process. The names of theses files are unknown at the time the user describes his simulation, thus they are considered as a parameter. This kind of parameter might be used to describe the future launching command of the job or used to replace patterns in some template text files. All the parameters are replaced by their effective values during the distribution process. Once the parameterization is finished, the application then automatically constructs the matching "Distributed Simulation" instance, representing a distributed version of the user's simulation. It is composed of common files needed for the simulation runtime and instances of the "SimulationJob" class, which contains all files generated from the "MetaFiles" and a literal launching command.

DistMe Processors

Simulations are considered as passive objects. The processor package, described in figure 4, contains the code to distribute and to prepare simulations for the runtime. The processors are useful to modify the simulation, to distribute it, to add literal launching commands and job description scripts (for a specified execution environment like OpenPBS or another job submission system). The "Distributor" class is able to create a "DistributedSimulation" from a "Simulation" object. The inherited classes from "Scripter" generate launching scripts for the execution environments. Extra execution environment support may be handled by adding specialized classes.



Figures 4: UML class diagram for Processors in DistMe



Figures 5: UML Representation of the Statuses in DistMe

Statuses and Serializers

DistMe has been designed to work with any targeted RNG library. Based on XML statuses the application provides an easy way to convert a status to a specific library format.

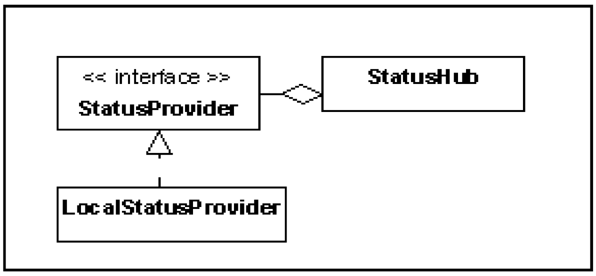
[1] http://sourceforge.net/projects/distme

Figure 4 shows the package used for the creation of status files. Instances of inherited class from the class called “Serializer” are able to generate a status file in a given specific library format using a “Status” object. At the moment, classes are implemented for very few different formats, but it is naturally extensible.

Figure 5 presents the RNG statuses as they are managed in DistMe. The “Status” class is common for all RNG statuses and contains meta-data about the RNG stream issued from the status. It is composed of an “RNGData” object, whose specialized classes contain effective data for RNG initialization. The “StatusSerializerXML” class contains generic code to instantiate any “Status” class instance from its XML description or to produce XML status files in a transversal manner from “Status” objects. The inherited classes from “StatusSerializerCLHEP” are translators to and from the CLHEP status file formats. Additional RNG status file format may be added by implementing inherited classes from the “Serializer” class.

The Status Distribution

The delicate part of an application is the distribution of the RNG statuses. The application is linked to a status database and selects the appropriate statuses. The distribution of RNG streams to each simulation job is made by the toolkit part named the “StatusProvider”. It can rely on a local database of statuses or use a web service. Figure 6, represents the stateless “StatusProvider” interface, which has been designed to be easily implementable by a web service. The web service is not yet operational; the application may rely on the “LocalStatusProvider”. The initialization of its database is presented in appendix A1. After this operation, the local status provider database is initialized in a persistent way on the local system. A same status may not be added twice in the database based on the SHA-1 hash code of the XML version of each status.

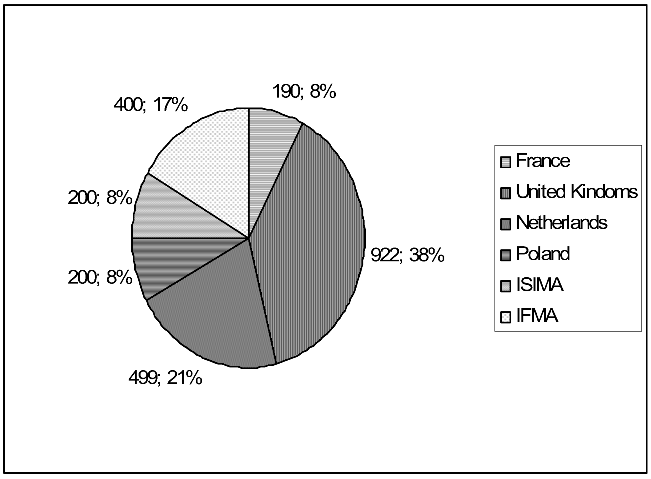


Figures 6: UML Representation of the StatusProvider in DistMe

The state of status distribution has been externalized in the “StatusHub” which ensures some constraints on returned statuses. It allows for example, in the case of a parallelization by sequence splitting the warranty of a minimum distance in term of number of drawings between each status used for the different jobs. It also ensures that a status is returned only once during a distribution process and guarantees by this mean the uniqueness of the random numbers streams. During the distribution process, RNG statuses are linked to “SimulationJobs” and the status files are created for a target RNG library.

APPLICATION

GATE (Jan S. et al. 2004) is a MCS tool based on the Geant4 package and dedicated to Single Photon Emission Computed Tomography and Positron Emission Tomography simulations. It was designed to be flexible and very precise, thus the price to pay is that GATE simulations are computationally intensive and cannot be directly used in a clinical context. DistMe has been applied to an image reconstruction application using GATE. For the execution part, we have used two local clusters and 650 worker nodes of a European grid environment (known as the Large Hardron Collider Grid). By distributing the calculation over many execution units our nuclear medicine simulation was achieved in a few days. It would have taken more than three years on a single powerful computer without distributing the simulation using the MRIP approach. Simulation results were directly used by scientists working in nuclear medicine (El Bitar et al. 2006; Breton and Buvat 2004; Lazaro et al. 2005). The distribution code may be found in our article appendix A2. DistMe is so fully usable for the distribution of stochastic simulation using the MRIP approach.



Figures 7: Repatition of simulation jobs execution

As shown in Figure 7, the execution of 2411 jobs needing 12 hours of execution each on recent computers (Pentium 4 3Ghz) has been executed using computing elements of 4 different countries and two clusters. 1811 jobs have been executed on the European computation grid and 600 on the clusters. In this massively parallel environment, the average execution time of a job was around 1 day; the total simulation would have taken 907 on a single computer, leading to an acceleration factor of more than 800.

To illustrate the generic aspect of DistMe the code to parallelize another code of spatial simulation of see grass colonization (Hill 1997) may be found in the appendix A3 of this article.

CONCLUSION

After the presentation of the theoretical basis on which is based the Distme toolkit, we have exposed its architecture and an example using a real simulation case study. At the time the authors are writing these lines, DistMe is fully operational to distribute stochastic simulations using the MRIP approach. Furthermore, this toolkit is already very extensible and features may be easily added with simple inheritance: new RNG status file format using the "Serializer" class, new RNG algorithms using "RNGData" class, new execution environments using the "Scripter" class and new sources for RNG statuses using the "StatusProvider" interface. Some parts of this architecture are about to evolve in a near future to take into consideration other distribution techniques, like the ones based on the generation of experiments (Amblard et al. 2003). A web service will be available and kept up to date, providing statuses and statistical tests results for the most commonly used RNG algorithm. Other parallelization techniques than the "sequence splitting" will be integrated and to be fully usable with DistMe. In parallel with this future work, a new random generation library, "DistRNG", will be developed with new features, like the initialization of generator from XML statuses, the integration of parallel statistical tests (Srinivasan et al. 1999) and new high quality RNG allowing fast cycle division (L'Ecuyer and Panneton 2005). Additional work will be done to address a reasonable level of fault tolerance in Grid environments and to include parameter exploration.

ACKNOWLEDGEMENTS

The authors would like to thank the Auvergrid Project, financed by the Auvergne Region and also the Engineering Schools IFMA and ISIMA Institutes which provided an easy access to their local clusters.

REFERENCES

Amblard F.; D.R.C. Hill.; S. Bernard; J. Truffot; and G. Deffuant. 2003. "MDA compliant Design of SimExplorer, A Software to handle simulation experimental frameworks", *Proceedings of SCSC 2003 Summer Simulation Conference* (Montréal, Jul. 20-24), 279-284.

Breton V. and I. Buvat. 2004. "Feasibility and value of fully 3D Monte Carlo reconstruction in Single Photon Emission Computed Tomography", *Nucl. Instr. and Meth. Phys. Res. A 527*, 195-200.

Coddington P.D. 1996. "Random number generator for parallel computers", *NHSE Review, 2nd issue*, Northeast Parallel Architecture Center.

Coddington P.D. and A.J. Newell. 2004. "JAPARA – A Java Parallel Random Number Library for High-Performance Computing", *Proceeding of the 18th International Parallel and Distributed Processing Symposium (IPDPS'04) - Workshop 5*, 156-166.

Darte A.; Robert Y. and Vivien F. 2000. Scheduling and Automatic Parallelization. Birkhauser, ISBN 0-8176-4149-1.

El Bitar Z.; D. Lazaro; V. Breton; D.R.C. Hill and I. Buvat. 2006. "Fully 3D Monte Carlo image reconstruction in SPECT using functional regions", *Nucl. Instr. Meth. Phys. Res*, in press.

Iskra K.A.; G.D. van Albada; and P.M.A. Sloot. 2005. "Towards Grid-Aware Time Warp", *Simulation: Transactions of The Society for Modeling and Simulation International 81*, 293-306.

Jan S. et al. 2004. "GATE: a simulation toolkit for PET and SPECT", *Phys. Med. Biol. 49*, 4543-4561.

Lazaro D.; Z. El Bitar; V. Breton; D. R. C. Hill; and I. Buvat. 2005. "Fully 3D Monte Carlo reconstruction in SPECT: a feasibility study", *Phys Med Biol 50*, 3739-3754.

L'Ecuyer P. and R. Simard. 2003. "TESTU01: a software library in ANSI C for empirical testing of random number generators", Manuscript, Department d'Informatique et de Recherche Operationnelle, University of Montreal, 1–206.

L'Ecuyer P. and F. Panneton. 2005. "Fast Random Number Generators Based on Linear Recurrences Modulo 2: Overview and Comparison", *Proceedings of the 2005 Winter Simulation Conference*, 110-119.

Lönnblad L. 1994. "CLHEP – a Project for designing a C++ Class Library for High Energy Physics", Computer Physics Communication 84, 307-316.

Maigne L.; D.R.C. Hill; P. Calvat; V. Breton; R. Reuillon; D. Lazaro; Y. Legre; and D. Donnarieix. 2004. "Parallelization of Monte Carlo Simulations and Sub-mission to a Grid Environment", *Parallel Processing Letters 14*, 177-196.

Mascagni M.; D. Ceperley; and A. Srinivasan. 2000. "SPRNG: A Scalable Library for Pseudorandom Number Generation", *ACM Transaction on Mathematical Software 26*, 618-619.

Mascagni M. and A. Srinivasan. 2004. "Parameterizing parallel multiplicative lagged-Fibonacci generators", *Parallel Computing 30*, 899-916.

Hill D.R.C. 1997. "Object-Oriented Pattern for Distributed Simulation of Large Scale Ecosystems", *SCS Summer Computer Simulation Conference* (Arlington, USA, Jul. 13-17), 945-950.

Matsumoto M. and T. Nishimura. 1997. "Mersenne Twister: A 623-dimensionally equidistributed uniform pseudorandom number generator", *Proceedings of the 29th conference on Winter simulation*, 127-134.

Matsumoto M. and T. Nishimura. 2000. "Dynamic Creation of Pseudorandom Number Generators", *Monte Carlo and Quasi-Monte Carlo Methods 1998*, 56-69.

Pawlikowski K. and V. Yau. 1992. "On Automatic Partitioning, Run-time Control and Output Analysis Methodology for Massively Parallel Simulations", *Proceedings of the European Simulation Symposium, ESS'92* (Dresden, Germany, SCS Int., Nov. 1992), 135-139.

Pawlikowski K. 2003. "Towards Credible and Fast Quantitative Stochastic Simulation", *Proceedings of International SCS Conference on Design, Analysis and Simulation of Distributed Systems, DASD'03* (Orlando, FL).

Rego V. J. and V. S. Sunderam. 1992. "Experiments in Concurrent Stochastic Simulation: the EcliPSe Paradigm", *Journal of Parallel and Distributed Computing 14*, 66-84.

Rycerz K.; M. Bubak; M. Malawski; and P.M.A. Sloot. 2005. "HLA Grid based support for simulation of vascular reconstruction", *Proceedings of the CoreGRID Workshop "Integrated Research in Grid Computin*g", 165-174.

Srinivasan A.; D. M. Ceperley; and M. Mascagni. 1999. "Random Number Generators for Parallel Applications", *Monte Carlo Methods in Chemical Physics 105*, 13-36.

Sunderam V. S. and V. J. Rego. 1991. "EcliPSe: A System for High Performance Concurrent Simulation", *Software Practice and Experience 21(11)*, 1189-1219.

Traore M. and D.R.C. Hill. 2001. "The use of random number generation for stochastic distributed simulation: application to ecological modeling", *proceedings of the 13th European Simulation Symposium* (Marseille, France, Oct. 18-20), 555-559.

Zeigler B.P.; H. Praehofer; and T.G. Kim. 2000. *Theory of Modelling and Simulation: 2nd Ed.: Integrating Discrete Event and Continuous Complex Dynamic Systems*, Academic Press.

BIOGRAPHY

ROMAIN REUILLON got his MSc in Computer Science in 2004 from ISIMA Computer Science & Modeling Institute (France). He obtained a Regional fellowship to begin his PhD studies in the Computer Science Laboratory at Blaise Pascal University (LIMOS). His research activities deal with stochastic simulations in distributed execution environments. (http://www.isima.fr/~reuillon)

DAVID R.C. HILL received his Ph.D. degree in Object-Oriented Simulation in 1993 from Blaise Pascal University. D. Hill is currently Full Professor and co-manager of the ISIMA Modeling Institute. His current application domain concerns Life Science Simulation. (www.isima.fr/hill)

APPENDIX

A1 Filling Up of the Local Status Provider

```
// Instantiate a Local Status Provider
p = new CLocalStatusProvider();

// Dir where the status are stored
// it could be a directory or an archive.
// The archive type is deduced from the extension,
// for more information go to truezip web site
dir = "./status.tar.gz/statusMT";

// The status files fit the 32 bits version
// of the Mersenne Twister of the CLHEP
// library format
deserializer =
            new CStatusSerializerCLHEPMT19937v32();

// Status file names are matching the following
// regular expression
filter = "MT[0-9]*\\.stat";

// The distance of a status is not mentioned into
// the file, it is so calculated using the
// status number included in the file name
// everything else is rejected.
 String [] reject = {"[a-z]","[A-Z]","\\."};

// The calculator use a part of the status name to
// deduce the distance (distance between two status
// is 15 billion drawings)
calculator =
   new CDistanceCalculator( reject,15000000000.0 );

p.populate( dir, deserializer ,
            filter, calculator );
```

A2 Parallelization of a Nuclear Medicine Application

Interpreted script (Java Like) used within the Distme toolkit

```
// set the work dir where files will be created
CEnvironment.GetInstance().setWorkDir
                       ( "/home/reuillon/tmp/work" );

//create the distributor for the simulation
distributor = new CDistributor();

//create the JobId parameter used during the
//distribution process
id = new CJobIdParameter( distributor );

//reference all the files needed for the
//simulator to work
inputFiles = new CFileReferences();
inputFiles.add( "GateMaterials.db" );
inputFiles.add( "Jaszczak.mac" );
inputFiles.add( "prerunGate.mac" );
inputFiles.add( "SourceJaszczak.mac" );
inputFiles.add( "ziadverbose.mac" );
inputFiles.add( "launchGate.sh" );

// set the file name format for RNG statuses
statusName = new CMetaFileName
                         ( "status", id, ".stat" );

//Create a status hub to extract status from a
//local database
hub = new CStatusHub( new CLocalStatusProvider(),
                      CRNGType.MT19937v32,
                      12000000000.0 );

//create a reference to the future RNG status file
statusRef = new CStatusFileReference(
          statusName,  hub,
          new CStatusSerializerCLHEPMT19937v32() );

//create a template file
macroName = new CMetaFileName( "macro",
                               id, ".mac" );
macroRef = new CTemplateFileReference
                          ( "Job.mac", macroName );

//replace some patterns in the files in a unique
//way for each simulation job

//add the replacement pattern for the RNG status
//name in the file
macroRef.addPattern( "RNGSTATUS", statusName );

//create a parameter matching the expected output
//string in the template file
output = new CConcatenatedParameter();
output.add( "output" );
output.add( id );

macroRef.addPattern( "ROOTOUTPUTFILE", output );
macroRef.addPattern( "BINOUTPUTFILE", output );

//reference the files
inputMetaFiles = new CMetaFileReferences();
inputMetaFiles.add( statusRef );
inputMetaFiles.add( macroRef );

//descibtion of the output files of each job
outputFiles = new CMetaFileReferences();
outputFiles.add(
  new CMetaFileReference( "output", id,".root" ) );
outputFiles.add(
   new CMetaFileReference( "output", id, ".bin" ));

//discribtion of the launching command
launchingCmd = new CLaunchingCommand(
                                  "launchGate.sh" );
launchingCmd.addArgument( macroName );

//describe the simulation
simulation = new CSimulationDescription(
                       inputFiles, inputMetaFiles,
                       launchingCmd, outputFiles );

//distribute the simulation
dist = distributor.distribute( simulation, 10 );

//create an object to generate the script for the
//future grid execution
CJDLScripter scripter = new
            CJDLScripter( "launchGate", ".jdl" ) );
```

```
scripter.setRequirements(
"(Member(\"VO-biomed-GATE-1.1.0-3\",other.GlueHostA
pplicationSoftwareRunTimeEnvironment)&&(other.GlueC
EPolicyMaxCPUTime>102))" );
scripter.process( dist );

//send all the file on a remote machine using ssh
//protocol
rm = new CRemoteMachine( "cclcgui.in2p3.fr" );
rm.selectTransfertProtocol( new CSSH
                                  ( "login",
                                    "password" ) );
rm.sendSimulation( dist, "/home/elbitar/romain" );
```

A3 Parallelization of an Environmental simulation (alga colonization)

```
CEnvironment.GetInstance().setWorkDir
                     ( "/home/reuillon/tmp/work" );

distributor = new CDistributor();
id = new CJobIdParameter( distributor );

inputFiles = new CFileReferences();
inputFiles.add( "simct" );
inputFiles.add( "mv.tca" );
inputFiles.add( "mv.tsu" );
inputFiles.add( "mv.tis" );
inputFiles.add( "mv.exp" );

statusName = new CMetaFileName
                          ( "status", id,".stat" );
hub = new CStatusHub( new CLocalStatusProvider(),
                      CRNGType.MT19937v32,
                      12000000000.0 );
statusRef = new CStatusFileReference
        ( statusName,  hub ,
          new CStatusSerializerCLHEPMT19937v32() );

inputMetaFiles = new CMetaFileReferences();
inputMetaFiles.add(statusRef);

outputFiles = new CMetaFileNames();

launchingCmd = new CLaunchingCommand( "./simct" );
launchingCmd.addArgument( "mv" );
launchingCmd.addArgument( statusName );

simulation = new CSimulationDescription
                     ( inputFiles, inputMetaFiles,
                       launchingCmd, outputFiles );

dist = distributor.distribute( simulation, 50 );

bash = new CBashScripter( "launch", ".sh" );
bash.setRedirectOutputs( true );
bash.process( dist );

CPBSScripter pbs = new CPBSScripter
                              ( "launch", ".pbs" );
pbs.setDefaultQueue( "q_these" );
pbs.process( dist );

rm = new CRemoteMachine( "remotemachine" );
rm.selectTransfertProtocol(
                 new CSSH( "login", "password" ) );
rm.sendSimulation( dist, "/remote/dir" );
```

SIMULATORS

AN APPROACH TO VIRTUAL-LAB IMPLEMENTATION USING MODELICA

Carla Martin, Alfonso Urquia and Sebastian Dormido
Dept. Informática y Automática, ETS Ingeniería Informática, UNED
Juan del Rosal 16, 28040 Madrid, Spain
E-mail: {carla, aurquia, sdormido}@dia.uned.es

KEYWORDS
Interactive simulation, object-oriented modeling, hybrid-models, education.

ABSTRACT

An approach to the implementation of virtual-labs is discussed in this manuscript. It allows describing the view (i.e., the user-to-model interface) and the model of the virtual-lab using Modelica language. To achieve this goal, the following two tasks have been completed: (1) a methodology to transform any Modelica model into a formulation suitable for interactive simulation has been proposed; and (2) *VirtualLabBuilder* Modelica library has been designed and programmed.

VirtualLabBuilder library includes Modelica models implementing a set of graphic interactive elements, such as containers, animated geometric shapes (polygon and oval) and interactive controls (slider and radio-button). These models allow the user: (1) to define the interactive graphic elements composing the virtual-lab view; and (2) to link the model variables with the geometric properties of these graphic elements.

The structure, capabilities and use of *VirtualLabBuilder* library are discussed in this manuscript. The library use is illustrated by means of a simple example. Finally, *VirtualLabBuilder* is used to implement the virtual-lab of the quadruple-tank process.

INTRODUCTION

A virtual-lab is a distributed environment of simulation and animation tools, intended to perform the interactive simulation of a mathematical model. Virtual-labs provide a flexible and user-friendly method to define the experiments performed on the model. In particular, interactive virtual-labs are effective pedagogical resources, well suited for distance education.

Typically, the virtual-lab definition includes the following two parts: the *model* and the *view*. The *view* is the user-to-model interface. It is intended to provide a visual representation of the model dynamic behavior and to facilitate the user's interactive actions on the model.

The graphical properties of the *view* elements are linked to the model variables, producing a bi-directional flow of information between the *view* and the *model*. Any change of a model variable value is automatically displayed by the *view*. Reciprocally, any user interaction with the *view* automatically modifies the value of the corresponding model variable.

Modelica (http://www.modelica.org/) is an object-oriented modeling language that facilitates the physical modeling paradigm (Åström et al. 1998). It supports a declarative (non-causal) description of the model, which permits better reuse of the models. As a consequence, the use of Modelica reduces considerably the modeling effort. However, neither Modelica language nor Modelica simulation environments (e.g., Dymola (Dynasim 2006)) support interactive simulation. As a consequence, extending Modelica capabilities in order to facilitate interactive simulation is an open research field.

Previous work on this topic includes (Engelson 2000; Martin et al. 2004; Martin et al. 2005a; Martin et al. 2005b). In particular:

- The combined use of Modelica/Dymola, Matlab and Easy Java Simulations (Esquembre 2004; http://www.um.es/fem/Ejs) is proposed in (Martin et al. 2004; Martin et al. 2005a; Martin et al. 2005b). This approach allows the implementation of virtual-labs with *runtime interactivity*. The user is allowed to perform actions on the model during the simulation run. He can change the value of the model inputs, parameters and state variables, perceiving instantly how these changes affect to the model dynamic. An arbitrary number of actions can be made on the model during a given simulation run.
- The combined use of Modelica/Dymola and Sysquake (http://www.calerga.com) is proposed in (Martin et al. 2005a). This approach facilitates the implementation of virtual-labs with *batch interactivity*. The user's action triggers the start of the simulation, which is run to completion. During the simulation run, the user is not allowed to interact with the model. Once the simulation run is finished, the results are displayed and a new user's action on the model is allowed.

The goal of the work discussed in this manuscript is the programming of a Modelica library supporting the implementation of virtual-labs with *runtime interactivity*. This novel Modelica library, named *VirtualLabBuilder*, allows the user to define the model and the view of the virtual-lab, and the link between them, using only Modelica.

The architecture and use of *VirtualLabBuilder* library is described in the following sections, and *VirtualLabBuilder* is used to implement the virtual-lab of the quadruple-tank process.

DESCRIPTION OF THE PROPOSED APPROACH

The virtual-lab description is composed of the model description and the view description.

a) The *virtual-lab model* has to be written in Modelica language, according to the methodology proposed in (Martin et al. 2005a). Essentially, this approach imposes that all the interactive variables have to be state variables. In particular, in order to allow interactive changes in the value of model parameters and input variables, they have to be written as zero-derivative state variables. This methodology can be applied to any Modelica model.

b) The Modelica description of the *virtual-lab view* (i.e., the view class) has to be a subclass of the `PartialView` Modelica class. `PartialView` is included in the *VirtualLabBuilder* library and it contains the code required to perform the model-to-view communication. This code is valid for any model and view descriptions, and the user only needs to set the length of the model-to-view communication interval.
In addition, the user has to include within the view class: (1) the required instantiations of the graphic interactive elements composing the virtual-lab view; and (2) the connection among these elements. The *VirtualLabBuilder* library contains a set of ready-to-use graphic elements. The connection among these elements determines their layout in the virtual-lab view. Dymola GUI allows defining in a drag-and-drop way the instantiation of these elements and connecting them using the mouse.

c) The Modelica description of the *virtual-lab* has to be an instance of `VirtualLab` class. This Modelica class is included in *VirtualLabBuilder* library. The user has to provide the name of the model class and the view class. Also, the user has to specify how the geometric properties of the view elements are linked to the model variables.

The virtual-lab description, obtained as discussed in c), is translated using Dymola and run. As a part of the model initialization (i.e., the calculations performed to find the initial value of the model variables), the initial sections of the interactive graphic objects and of the `PartialView` class are executed. These initial sections contain calls to Modelica functions, which encapsulate calls to external C-functions. These C-functions are Java-code generators.

As a result, during the model initialization, the Java code of the virtual-lab view is automatically generated, compiled and bundled into a single jar file. Also, the communication procedure between the model and the view is set up. This communication is based on client-server architecture: the C-program generated by Dymola is the server and the Java program automatically generated during the model initialization is the client.

Once the jar file has been created, it has to be executed by the user. As a result, the initial layout of the virtual-lab view is displayed and the client-server communication is established. Then, the model simulation starts.

During the simulation run, there is a bi-directional flow of information between the model and the view. The communication is as follows. Every communication interval:

- The model simulation (i.e., the server) sends to the view (i.e., the client) the data required to refresh the view.
- The view sends to the model simulation the new value of the variables modified due to the user's interactive action.

VirtualLabBuilder ARCHITECTURE

VirtualLabBuilder library is composed of the following five packages (see Figure 1a).

- The `ViewModel` package contains the `PartialView` and the `VirtualLab` classes.
- The `ViewElements` package contains the graphic interactive elements that the user can employ to compose the view. The content of this package is shown in Figure 1b and it will be described in the next section.
- The `Interfaces` package contains the interfaces (i.e., connectors) of the graphic interactive elements.
- The `Functions` package contains the Modelica functions which encapsulate calls to external C-functions. As discussed in the previous section, these C-functions are Java-code generators.
- The `TypesDef` package contains the definition of several types of variables. These types are intended to be used for defining some properties of the graphic interactive elements (such as color, layout, etc.).

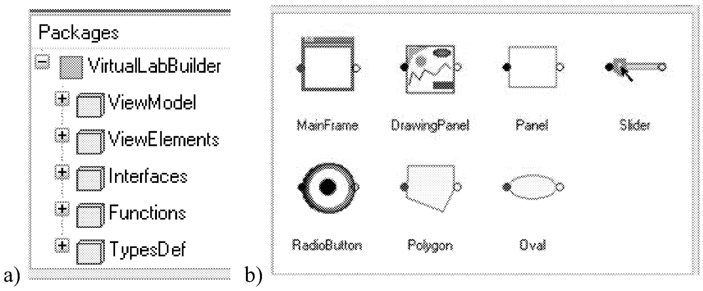


Figure 1: a) Packages of the *VirtualLabBuilder* Library; and b) Classes within the `ViewElements` Package

GRAPHIC ELEMENTS

The `ViewElements` package contains the graphic elements that can be used to define the view. These elements (see Figure 1b) can be classified into the following three categories.

- *Containers* (`MainFrame`, `Panel` and `DrawingPanel` classes). These graphic elements that can host other graphic elements. The properties of these elements are set in the view definition and they can not be modified during the simulation run.

- *Drawables* (`Polygon` and `Oval` classes). These elements can be used to build an animated schematic representation of the system. The variables setting the geometric properties of these elements (position, size, etc.) can be linked to model variables.
- *Interactive controls* (`Slider` and `RadioButton` classes). Model variables can be linked to the variables defining the states of the interactive control elements. This allows the user to change the value of these model variables during the simulation run.

Drawable elements and interactive controls implement the information flow between the model and the view of the virtual-lab. The simulated value of the model variables modifies the properties of the drawable elements (i.e., model-to-view information flow). The user's interactive action on the interactive controls modifies the value of the model variables (i.e., view-to-model information flow). The properties of the graphic elements are discussed next.

Containers

`MainFrame` class creates a window where containers and interactive controls can be placed. The view can only contain one `MainFrame` object. This class has the following parameters:

- *width* and *height*: width and height of the window in pixels.
- *title*: text shown in the top part of the window.
- *layoutPolicy*: layout policy of the element. It sets where the elements placed within the window are located. Possible values are BorderLayout, GridLayout and FlowLayout.

`Panel` class creates a panel where containers and interactive controls can be placed. This component is similar to `MainFrame`, however there is a difference: the view can contain more than one panel.

`DrawingPanel` class creates a two-dimensional container that only can contain drawable objects (i.e., `Polygon` and `Oval` objects). It represents a rectangular region of the plane which is defined by means of two points: (*XMin*, *YMin*) and (*Xmax*, *YMax*). The coordinates of these two points (i.e., the value of *XMin*, *XMax*, *YMin* and *YMax*) are parameters of the class whose value can be set by the user.

Drawables

`Oval` class draws an oval. The position and the lengths of the axes can be linked to the model variables. The class has the following parameters:

- *lineColor*: color of the line.
- *fillColor*: color of the filling.
- *filled*: indicates whether the oval is filled or empty.
- *intCenter*: indicates whether the oval position changes during the simulation or remains constant.
- *intAxes*: indicates whether the oval shape changes during the simulation or remains constant.

`Polygon` class draws a polygonal curve specified by the coordinates of its vertexes points. The class has the following parameters:

- *lineColor*: color of the line.
- *fillColor*: color of the filling.
- *filled*: indicates whether the oval is filled or empty.
- *intVertexesX*[:]: array that indicates whether the horizontal position of each polygon's point changes during the simulation or remains constant.
- *intVertexesY*[:]: array that indicates whether the vertical position of each polygon's point changes during the simulation or remains constant.

Interactive Controls

`Slider` class creates a slider. This class has the following parameters:

- *position*: slider position inside the container object.
- *stringFormat*: format used to display the value.
- *tickNumber*: number of ticks.
- *tickFormat*: tick format.
- *enable*: allows enabling/disabling the object.
- *initialValue*: initial value of the slider variable.

`RadioButton` class creates a radio button control.

CONNECTING THE GRAPHIC ELEMENTS

The interfaces of the *container*, the *drawable* and the *interactive control* classes are composed of two connectors: one filled and one empty (see Figure 1b). The user must observe the following three rules when connecting the graphic elements:

1. The connection between two components must be established by connecting the filled connector of one component with the empty connector of the other component.
2. Each filled connector must be connected to one and only one empty connector.
3. Empty connectors can be left unconnected. If they are connected, the allowed number of filled connectors connected to a given empty connector depends on the type of the graphic elements. This number is shown in Figure 2.

		filled connector				
		MainFrame	DrawingPanel	Panel	Interactive Controls	Drawable
empty connector	MainFrame		≥1$^{(*)}$	≥1$^{(*)}$	≥1$^{(*)}$	
	DrawingPanel					1
	Panel		≥1$^{(*)}$	≥1$^{(*)}$	≥1$^{(*)}$	
	Interactive Controls				1	
	Drawable					1

Figure 2: Allowed Number of Connections

(*): If the layout policy of the element is BorderLayout then ≥1 else 1.

LIBRARY USE

The steps to compose the Modelica description of a virtual-lab are described below. They are illustrated by means of a simple example: the implementation of the virtual-lab of the tank model shown in Figure 3.

*Model definition***.** The voltage applied to the pump (v) is an input variable. The cross-sections of the tank (A) and the outlet hole (a), the pump parameter (k) and the gravitational acceleration (g) are time-independent properties of the physical system. The physical parameters A and a, and the input variable v can be modified by the user action during the interactive simulation. Interactive parameters and input variables have been declared in the model as Real variables with zero time-derivative. It is assumed that the Modelica class of the tank has been programmed and it is called `PhysicalModel`.

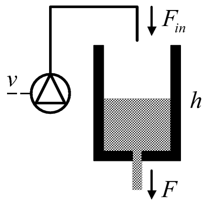
View definition. Create a new class extending the `PartialView` class and call it `ViewModel` Set the value of the model-to-view communication interval, which is a parameter (called `Tcom`) of the `PartialView` class. The `PartialView` class contains one graphic element: `root`. The object of the `MainFrame` class must be connected to this element. Add the `MainFrame` object and the other required graphic objects to the `ViewModel` class. Connect the graphic objects. The diagram of the obtained class is shown in Figure 4. Set the value of the graphic object parameters.

Virtual-lab definition. Create a new object of the `virtualLab` class. This class contains two parameters: the class of view (`ViewI`) and the class of the model (`ModelI`). Set the values of these parameters to `ViewModel` and `PhysicalModel` respectively. Finally, write the equations required to link the view variables with the model variables.

Virtual-lab run. Translate (for instance, using Dymola) and simulate the object created previously. During the initialization calculations, the jar file is automatically generated. Execute the jar file. The virtual-lab view is displayed (see Figure 5a) and the interactive simulation of the virtual-lab starts. The time evolution of the model variables can be plot using Dymola. It is shown in Figure 5b for some selected variables. The discontinuous changes are due to user's interactive actions.

CASE STUDY: IMPLEMENTATION OF THE QUADRUPLE-TANK PROCESS VIRTUAL-LAB

The quadruple-tank process is represented in Figure 6. It can be used to explain different aspects of the multivariable control theory (Johansson 2000; Dormido and Esquembre 2003). The goal is to control the level of the two lower tanks with two pumps. The virtual-lab has been implemented as described in the previous section. It supports interactive changes in the liquid levels, the pump input voltages and the valve settings. The Modelica description of the view is shown in Figure 7 and the virtual-lab view in Figure 8. The time evolution of the liquid levels can be plot using Dymola (see Figure 9).



$$\frac{dV}{dt} = F_{in} - F$$
$$F = a\sqrt{2gh}$$
$$V = Ah$$
$$F_{in} = kv$$

V: liquid volume
h: liquid level
a: hole cross-section
A: tank cross-section
F, F_{in}: liquid flow
g: grav. acceleration
v: pump input voltage
k: pump parameter

Figure 3: Model of a Tank System

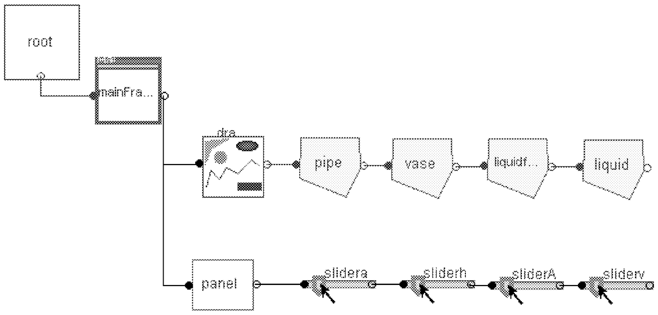


Figure 4: Diagram of the Modelica Description of the View

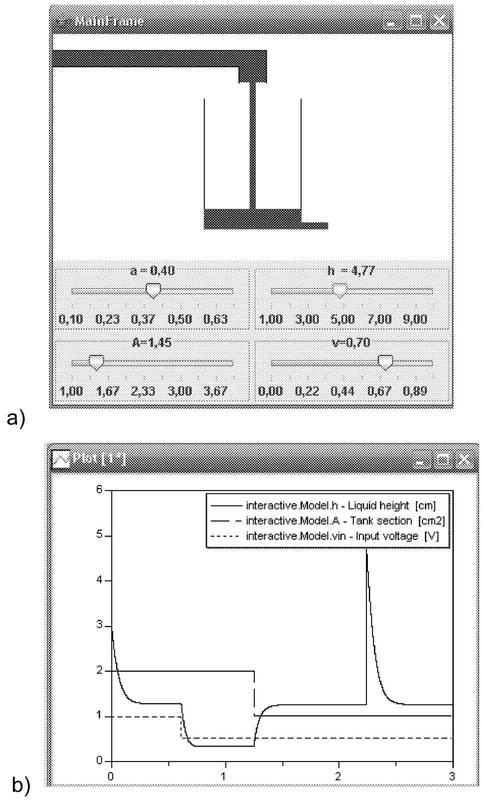


Figure 5: a) Virtual-lab View; b) Variable Plots

CONCLUSIONS

A novel approach to the virtual-lab implementation using Modelica language has been proposed. In order to put it into practice, two tasks have been completed: (1) the proposal of a modeling methodology intended to transform any Modelica model into a description suitable for interactive simulation; and (2) the design and programming of a Modelica library supporting the description of the virtual-lab view and the bi-directional communication between the model and the view. The purpose, structure and use of this Modelica library, called *VirtualLabBuilder*, have been discussed and its use has been illustrated by means of two examples.

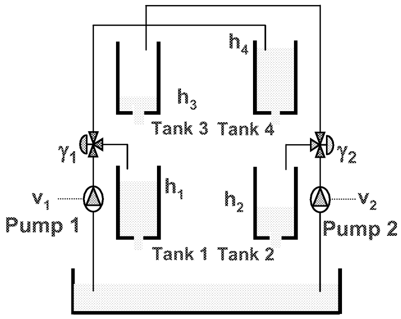


Figure 6: The Quadruple-Tank Process

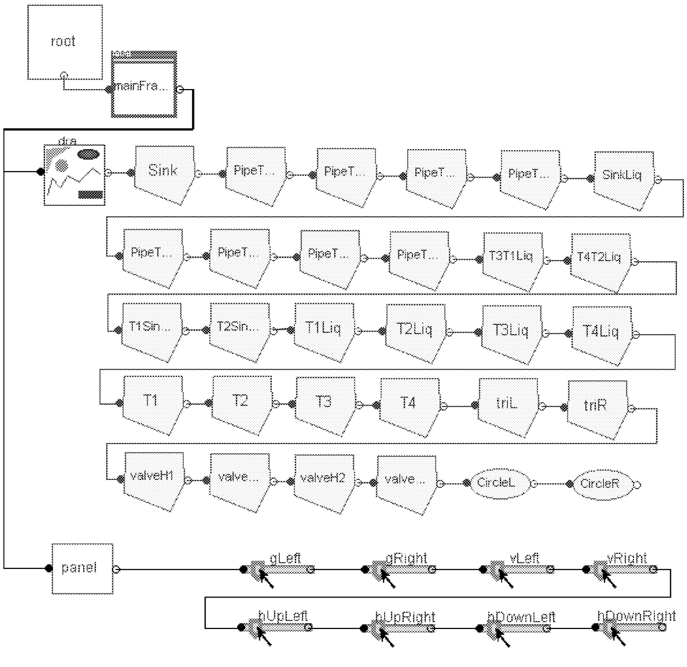
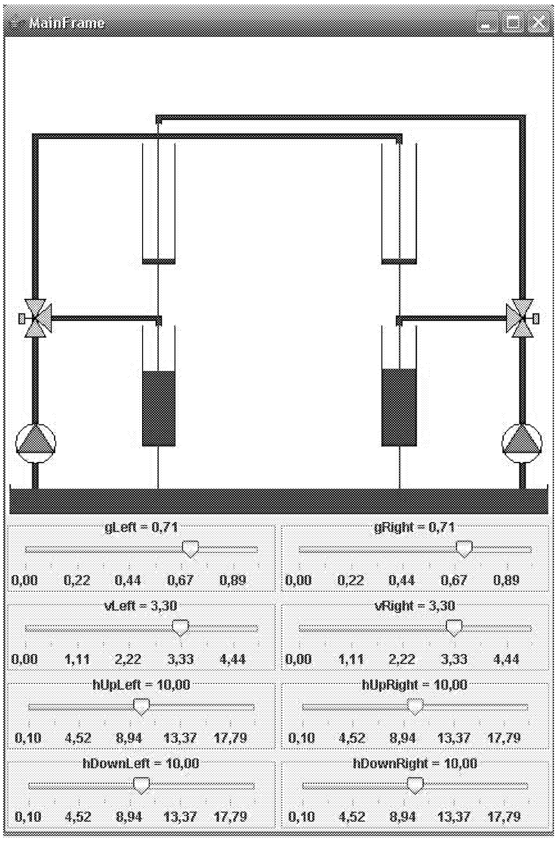


Figure 7: Diagram of the Modelica Description of the View



gLeft = 0,71

0,00 0,22 0,44 0,67 0,89

gRight = 0,71

0,00 0,22 0,44 0,67 0,89

vLeft = 3,30

0,00 1,11 2,22 3,33 4,44

vRight = 3,30

0,00 1,11 2,22 3,33 4,44

hUpLeft = 10,00

0,10 4,52 8,94 13,37 17,79

hUpRight = 10,00

0,10 4,52 8,94 13,37 17,79

hDownLeft = 10,00

0,10 4,52 8,94 13,37 17,79

hDownRight = 10,00

0,10 4,52 8,94 13,37 17,79

Figure 8: Virtual-Lab View

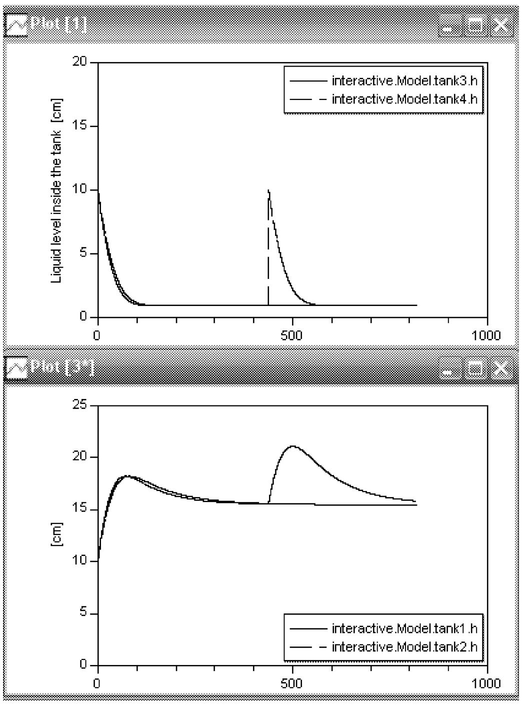


Figure 9: Plots of Selected Process Variables

REFERENCES

Åström K.; H. Elmqvist and S. E. Mattsson. 1998. “Evolution of Continuous-Time Modeling and Simulation”. In *Proceedings of the 12th European Simulation Multiconference* (Manchester, UK).

Dormido, S. and F. Esquembre. 2003. "The Quadruple-Tank Process: An Interactive Tool for Control Education", *In Proceedings of the 2003 European Control Conference*, (Cambridge, UK).

Dynasim. 2006. “Dymola. User’s Manual”. Dynasim AB. Lund, Sweden.

Engelson V. 2000. “Tools for Design, Interactive Simulation, and Visualization of Object-Oriented Models in Scientific Computing”. Ph. D. Thesis, Dept. of Computer and Information Science, Linköping University, Sweden. Dissertation No. 627.

Esquembre F. 2004. “Easy Java Simulations: a Software Tool to Create Scientific Simulations in Java”. *In Computer Physics Communications*, Vol. 156, 199-204.

Fritzson P. 2004. “Principles of Object-Oriented Modeling and Simulation with Modelica 2.1”. John Wiley & Sons.

Johansson K.H. 2000. “The Quadruple-Tank Process: A Multivariable Laboratory Process with an Adjustable Zero”, *IEEE Transactions on Control Systems Technology*, Vol. 8, No. 3 (May), 456-465.

Martin C; A. Urquia; J. Sanchez; S. Dormido; F. Esquembre; J.L. Guzman and M. Berenguel. 2004 “Interactive Simulation of Object-Oriented Hybrid Models, by Combined use of Ejs, Matlab/Simulink and Modelica/Dymola”, *In Proc. of the 18th European Simulation Multiconference*, 210-215 (Magdeburg, Germany).

Martin C.; A. Urquia and S. Dormido. 2005a. “Object-oriented modelling of interactive virtual laboratories with Modelica”. *In Proc. of the 4th Int. Modelica Conference*, 159-168 (Hamburg, Germany).

Martin C.; A. Urquia and S. Dormido. 2005b. “Object-oriented modeling of virtual laboratories for control education”. *In Proc. of the16th IFAC World Congress*, paper code Th-A22-TO/2 (Prague, Czech Republic).

AN INTEGRATED VEHICULAR AND NETWORK SIMULATOR FOR VEHICULAR AD-HOC NETWORKS

Cristian Gorgorin, Victor Gradinescu,
Raluca Diaconescu and Valentin Cristea
"Politehnica" University Bucharest
Computer Science Department
313 Splaiul Independentei Bucharest Romania
E-mail: {cristig, victor, ralucad}@egov.pub.ro,
valentin@cs.pub.ro

Liviu Iftode
Rutgers University
Computer Science Department
110 Frelinghuysen Road
Piscataway, New Jersey, USA
E-mail: iftode@cs.rutgers.edu

KEYWORDS

Intelligent Transportation Systems, Vehicular Ad-Hoc Networks, simulation, discrete-event, integration, vehicular mobility, wireless communication, network

ABSTRACT

Vehicular ad-hoc networks (VANETs) form when vehicles are equipped with devices capable of short-range wireless communication. Accurate simulation of VANETs is a challenging task, requiring both a vehicle mobility model and a network simulator. Although separate simulators exist, integrating them is difficult. We have developed an integrated simulator, based on studied, validated models. We argue that our simulator can be used for the studying of a large range of VANET protocols and applications, which would be very difficult to study by using other tools.

INTRODUCTION

Vehicle-to-vehicle communication is a very challenging topic in recent years. Vehicles equipped with devices capable of short-range wireless connectivity can form a particular mobile ad-hoc network, called a "Vehicular Ad-hoc NETwork" (VANET). The existence of such networks opens the way for a large range of applications. We consider that two of the most important classes of such applications are those related to route planning and traffic safety.

Route planning aims to provide drivers with real-time traffic information, which would, in the absence of a VANET, require expensive infrastructure. By contrast, the VANET approach is highly scalable and has very low maintenance costs. Moreover, short-range wireless communication technologies (such as 802.11) have no associated cost, other than the communication device.

Safety applications involve disseminating urgent information, which is unavailable in the driver's field of view, or is difficult to notice for reasons such as fog or other vehicles obstructing the line of sight. For instance, a lot of accidents occur in foggy conditions, because drivers notice too late that some kind of incident has occurred in front of them. Safety at intersections could also be enhanced, because the risk of collisions could be detected in advance and the driver could be warned seconds before what would otherwise be an imminent accident.

Most applications to be deployed on top of a VANET require some sort of data-dissemination model. This is a challenging problem, due to the unique characteristics of a VANET. Such a network has a very high degree of nodes' mobility and a very large scale. Network partitioning occurs frequently, making end-to-end communication impossible at times. Several studies (Blum et.al. 2004) show that the performance of classical, topology-based routing protocols in vehicular networks is poor, due to the extremely high mobility of the nodes.

The evaluation of VANET protocols and applications could be made through real outdoor experiments, which should involve a large number of nodes, in order to obtain significant results. However, performing such large-scale experiments is extremely difficult. Therefore, simulation is an indispensable tool.

The simulation of a VANET requires two different components: a network simulator, capable of simulating the behavior of a wireless network, and a vehicular traffic simulator, able to provide an accurate mobility model for the nodes of a VANET. Recent studies (Choffnes and Bustamante 2005) have proven that the vehicular mobility model is very important, and in order to obtain relevant results, it should be well integrated with the wireless network model. The use of an inaccurate mobility model, like the popular random waypoint model (which may work for some mobile ad-hoc networks, but is definitely not an accurate representation of mobility in a VANET), can lead to erroneous results (Choffnes and Bustamante 2005) (Saha and Johnson 2004).

We have developed a simulation tool, comprising the two previously mentioned components: a microscopic traffic simulator, and a wireless communication model. We have also implemented a graphical user interface, using OpenGL for Java (JOGL), which proved useful in some phases of the simulation experiments, but which can be disabled in order to shorten the simulation time.

The remainder of the paper is organized as follows. In the following section we present related work in the area of vehicular networks simulators, along with the motivation for developing our integrated simulator. Another section presents the simulator we have developed. We then show evaluation results and we briefly present applications which could be studied using our simulator. We conclude in the final section.

RELATED WORK

Simulating a vehicular network involves two different aspects. First, there are issues related to the network, such as medium access control, signal strength, propagation delays. Network simulators, like "The Network Simulator – ns-2" (http://www.isi.edu/nsnam/ns) and Jist/SWANS (http://jist.ece.cornell.edu/index.html), cope with such issues. However, a general-purpose wireless network simulator is by no means enough for an accurate simulation of a vehicular network. Nodes in a wireless network usually move according to the random-waypoint model. This means they have an origin and a destination and move towards the destination. But vehicles only move along roads and that is a very particular situation. Furthermore, real vehicles move according to very particular traffic models, due to the street topology, intersections, traffic regulations and drivers' behavior. That takes us to the second very important aspect of a vehicular network simulator, which is using a mobility model as close as possible to real vehicular mobility.

Vehicular traffic simulators can be classified in macroscopic and microscopic simulators. Macroscopic simulators deal with global measures, like traffic flow, while microscopic simulators take into account the movement of each particular vehicle.

There are a lot of commercial vehicular traffic simulators. They have not been designed especially for vehicular computing. They are primarily used to study traffic, in order to validate projects, like building a new road, or a new tram line, or for designing effective traffic signals.

An example of a commercial vehicular traffic simulator is VISSIM (http: // www.english.ptv.de / cgi-bin / traffic/traf_vissim.pl). It is a microscopic simulator and implements driver behavior models, like car-following or lane changing. According to its producers, it is used in over 70 countries. An integrated simulator was developed by a team at Northwestern University. It is based on an original vehicular traffic model, called Street Random Waypoint (STRAW). Their simulator is implemented on top of JiST/SWANS, and it is free and open-source (Choffnes and Bustamante 2005). The authors have used the simulator in order to prove that studying routing protocols for a vehicular network without an accurate vehicular traffic model is a wrong approach. In this respect, they compared results obtained with the Random Waypoint model (which is a very inaccurate representation of a vehicular network) with results obtained with the STRAW model. Their experiments clearly indicate that using the Random Waypoint model will not produce accurate results for a vehicular network.

However, we believe the mobility model implemented in existing simulators (Choffnes and Bustamante 2005) (Saha and Johnson 2004) (Mangharam et.al. 2005) is not a sufficiently accurate representation of real vehicle mobility. Thus, the simulator of Saha and Johnson uses real maps, in the TIGER format (*http://www.census.gov/geo/www/tiger*) and vehicles move along the streets. Each vehicle moves completely independent of other vehicles, with a constant speed randomly chosen. Multi-lane roads or traffic control systems are not taken into consideration. Other authors (Mangharam et.al. 2005) make the same oversimplifying assumptions and do not consider multi-lane roads or car-following models. The mobility model of Choffnes and Bustamante is more complex. It also uses TIGER files, and considers car-following models. The motion of a vehicle is influenced by the preceding vehicle. The authors also implement traffic control systems: timed traffic lights and stop signs. However, multi-lane roads are not taken into consideration.

Furthermore, the majority of VANET applications imply that vehicles react to messages. For instance, if a driver receives a message saying that the road ahead is congested, that driver will change its route. In order to study such reactions, combining an existing vehicular traffic simulator with an existing wireless network simulator is not possible. An integrated simulator is needed.

Based on these aspects, we have chosen to develop a VANET simulation tool, integrating vehicular mobility and wireless transmission simulator.

DESCRIPTION OF THE SIMULATOR

Architecture

The VANET simulator we have developed is a discrete event simulator. The simulation time advances with a fixed time resolution after executing the application code for the current simulation time. More specifically, at every moment of the simulation time, all the current events are pulled from a queue of events, and handled in a random order.

The events queue can hold three types of events: *send*, *receive* or *GPS*. A *send* event for a specified node triggers the calling of the node's procedure responsible for preparing a message. It also schedules the corresponding *receive* event(s) for the receiver(s) the simulator decides to deliver the message to, according to the network module. The *receive* event is associated either with a node, or with a group of nodes (to which the message is broadcasted). Its action is to call the appropriate handler in each of the receiving nodes. The *GPS* event is scheduled at a regular time interval for each node, in order to simulate the way a real VANET application collects GPS data periodically.

The mobility module updates periodically the position of each node representing a vehicle, according to the vehicular mobility model. This model takes into account vehicle interactions (passing by, car following patterns etc), traffic rules and the behavior of different drivers.

The main advantage of this architecture is that the simulator can execute (or emulate) the code of a real vehicular application without significant changes, by using the interface described above. Figure 1 shows the general view of this simulation environment.

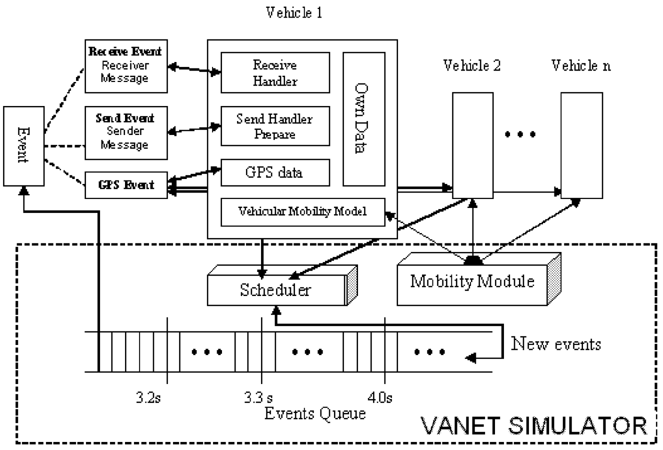


Figure 1 : Simulator architecture

Mobility Model

Maps

A digital map is required in any kind of VANET application. Each vehicle which is part of the system should have such a digital map. For our simulator, we have chosen to use TIGER files, which are freely-available, real digital maps of the USA (http://www.census.gov/geo/www/tiger). The TIGER files contain detailed geographical information about all the roads in a region, from large highways to small streets. The data they contain come in the form of geographical coordinates (latitude, longitude) for the roads. Thus, for every road, the TIGER files specify its end points, along with as many intermediary points as needed, depending on the road's shape. Furthermore, for each road, a "class" information is given (whether it is a small street, a local road, a State Route, an Interstate Highway and so on).

However, the TIGER database unfortunately lacks other traffic-specific information, like the number of lanes, or traffic control systems (traffic lights, yield or stop signs). We believe that a mobility model which does not take multiple lanes or traffic control systems into consideration is not realistic enough; therefore we have added some extra information, based on simple heuristics and based on the road class information included in the TIGER files. Some of the rules we have used include more lanes for higher class roads, yield or stop signs for lower class roads, traffic lights between equally important roads, longer green period for the road with the higher number of lanes and so on. In the future, we can probably expect such detailed traffic information to be contained in real digital maps.

Microscopic Traffic Simulator

A traffic simulator which takes into account the actions of each individual vehicle is a microscopic simulator, as opposed to macroscopic simulators, which describe the evolution of traffic using global measures, like flow or traffic density. Macroscopic simulators can be used to better understand the traffic dynamics and to better design traffic-related facilities (traffic lights, number of lanes, lane closures and so on). However, a much higher level of detail is necessary for the study of a vehicular network; therefore we have developed a microscopic traffic simulator. It is based on the driver behavior model developed by Wiedemann (1974, 1991). The same model is used in the commercial traffic simulator "VISSIM". Like many other vehicular traffic simulators, VISSIM's purpose is modeling and forecasting vehicle traffic flow, for decisions like adding a new lane, studying the impact of lane closures on traffic, building an overpass and so on. Such simulators are difficult to integrate with network simulators, especially since most of them are commercial products.

Next, we briefly describe the driver behavior model we have implemented, which is based on the idea developed by Wiedemann, and further studied by Fellendorf and Vortisch (2000). The basis assumption is that a driver can be in one of four modes: free driving, approaching, following or braking.

Free driving means there is no influence from preceding vehicles on the same lane. In this situation, the driver will seek to obtain and maintain a desired speed. The desired speed and the acceleration depend on the driver personality, and on the road characteristics.

In the "approaching" mode there is a slower, preceding vehicle that influences the driver. In this situation, she/he will apply a deceleration in order to obtain the same speed as the preceding vehicle. The deceleration is a function of the distance between the two vehicles, their speeds, as well as other parameters.

The "following" mode means there is a preceding vehicle, but the speeds of the two vehicles are practically equal. In this situation, the driver will seek to keep the speed constant.

The "braking" mode means there is a slower preceding vehicle, very close in front. In this mode, due to the immediate danger, the driver will apply high deceleration rates.

Figure 2 presents some basic rules to determine the mode that corresponds to a driver. Thus, there are two thresholds, "distance1" and "distance2" according to the notation in the figure. If the preceding vehicle is closer than "distance1" and slower than the current vehicle, then the latter will be in "braking mode". If the slower, preceding vehicle is between "distance1" and "distance2" in front, then the mode will be "approaching", and the current vehicle will gradually decelerate. If the preceding vehicle is further away than "distance2", then it does not influence the current vehicle in any way, and it will be in the "free driving" mode. These thresholds ("distance1" and "distance2") are not constant, but they depend on the driver's personality and on the vehicle's speed.

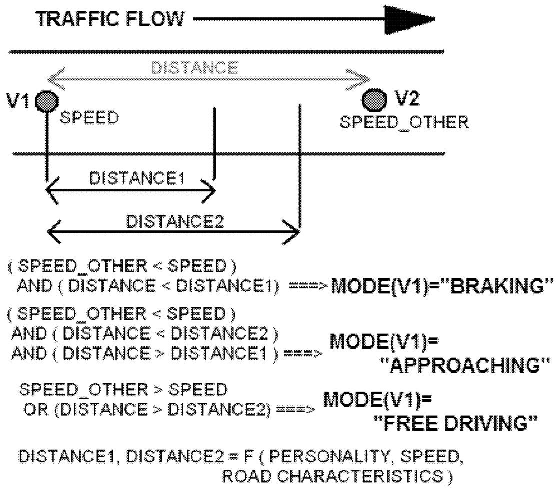

TRAFFIC FLOW
DISTANCE
V1
SPEED
V2
SPEED_OTHER
DISTANCE1
DISTANCE2
(SPEED_OTHER < SPEED)
AND (DISTANCE < DISTANCE1) ===> MODE(V1)="BRAKING"
(SPEED_OTHER < SPEED)
AND (DISTANCE < DISTANCE2)
AND (DISTANCE > DISTANCE1) ===> MODE(V1)= "APPROACHING"
SPEED_OTHER > SPEED
OR (DISTANCE > DISTANCE2) ===> MODE(V1)= "FREE DRIVING"
DISTANCE1, DISTANCE2 = F (PERSONALITY, SPEED, ROAD CHARACTERISTICS)

Figure 2 : Driver modes

We have also implemented a lane-changing model, for multi-lane roads. The model we have implemented is based on the lane-usage rules valid throughout most part of Europe. Thus, the usage of the first lane is required, unless it is occupied. It means that a driver will always try to stay on the lower lanes, except when overtaking another slower vehicle. Overtaking on the right side is not allowed. These rules are not valid in city environments, near intersections, where lanes are selected based on the direction the driver intends to follow.

The lane-changing model we have designed and implemented is based on a hierarchy between the four driving modes. Whenever a driver is in a different mode than "free driving", she/he will always check if the higher lane can provide a superior mode. If that is the case, the driver will switch to a higher lane. Similarly, whenever a driver is in a different mode than "braking", she/he will always check if the lower lane provides at least similar conditions. If that is the case, the driver will switch to a lower lane. The order of these checks is important. The higher lane is checked first.

Thus, if a driver uses lane 2 and approaches another slower vehicle, it will first check if lane 3 is empty, and if that is the case it will switch to lane 3 (only if it can safely complete the switch, without interfering with any vehicles approaching from behind). If it had first checked the lower lane, it could have discovered that it is empty and it would have decided to use lane 1 for overtaking the vehicle on lane 2, which is forbidden in most European countries. However, it is not forbidden in the United States, where any lane can be used for overtaking. US traffic could easily be simulated, by making a random decision whether to first check the higher lane or the upper lane when looking for superior driving conditions.

We have also incorporated traffic control systems into our driver behavior model implementation. Thus, the vehicles we simulate are aware of traffic lights, priority roads and "yield" or "stop" signs, and their motion is simulated according to these traffic control systems.

Different driver profiles (aggressive, regular, calm) can easily be modeled by using the numerous model parameters. Each driver class is represented by a certain set of values for the parameters. In order to further differentiate the drivers, there is also a small deviation from the specified values, deviation computed randomly for each driver.

Fellendorf and Vortisch (2000) proved that the model is accurate, by comparing simulated traces with real measurement data taken from a German freeway and from a US freeway. Still, the model is supposed to be accurate not only for freeway conditions, but also for city-like scenarios. To further calibrate and validate our model, we have focused on a simple, yet very frequent, city-like scenario. We considered a typical intersection where vehicles are queued, waiting for a red traffic light to become green (see Figure 3a). We assumed that all vehicles intend to drive forward. Let "FlowPerLane" be the number of vehicles that pass the intersection per second, per lane, during a time period beginning immediately after the light has become green. We consider "FlowPerLane" to be a very important parameter characterizing the motion of vehicles through the intersection, because it is influenced by several parameters of our driver behavior model, like vehicle acceleration, desired distance from the preceding vehicle, or reaction time.

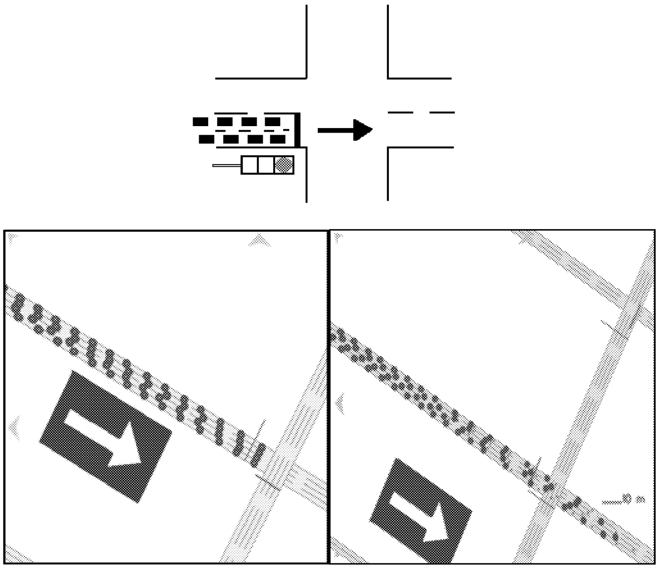


Figure 3 : Typical city scenario and two simulation screenshots

We have chosen the intersections "Piata Victoriei" and "Arcul de Triumf" in downtown Bucharest for measurements. Both intersections meet the above mentioned assumptions. The number of lanes is large (4, respectively 3), and all vehicles are required to drive forward. We measured FlowPerLane by counting passing vehicles during the green phase of the traffic light. We repeated the experiment several times, varying the time frame during which we counted the vehicles, because we suspected there might be a difference between the flow values at the beginning and towards the end of the green phase. The results, however, did not indicate such a difference. We simulated a similar intersection using our driver behavior model. Figure 3 shows screenshots of our simulator's JOGL GUI, taken during the simulation. Figure 3b shows the vehicles still waiting for the red light to become green, while Figure 3c shows the vehicles as they have started passing, as the light has turned green. We have calibrated some of the numerous driver behavior model parameters, based on the real results obtained. Finally, with the calibrated parameters, we have performed several measurements. The measured data and the simulated data are presented in the table in Figure 4.

It is easy to see the similarity between the simulated data and the real situation. The simulated data values have an average of **0.46** and a standard deviation of *0.03*. The measured data values from "Piata Victoriei" have an average of **0.45** and a standard deviation of *0.04*. Finally, the measured data values from "Arcul de Triumf" have an average of **0.47** and a standard deviation of *0.02*.

SIMULATED DATA				REAL MEASURED DATA			
NUMBER OF VEHICLES	NUMBER OF LANES	TIME (SECONDS)	FLOW PER LANE (VEHICLES/SEC)	NUMBER OF VEHICLES	NUMBER OF LANES	TIME (SECONDS)	FLOW PER LANE (VEHICLES/SEC)
47	4	26	**0.45**	PIATA VICTORIEI			
46	4	26	**0.44**	12	1	26	**0.46**
49	4	26	**0.47**	12	1	26	**0.46**
49	4	26	**0.47**	10	1	26	**0.38**
48	4	26	**0.46**	49	4	26	**0.47**
51	4	26	**0.49**	50	4	26	**0.48**
47	4	26	**0.45**	6	1	12	**0.50**
48	4	26	**0.46**	18	4	12	**0.38**
25	4	12	**0.52**	23	4	12	**0.48**
24	4	12	**0.50**	22	4	12	**0.46**
22	4	12	**0.46**	ARCUL DE TRIUMF			
24	4	12	**0.50**	58	3	40	**0.48**
24	4	12	**0.50**	55	3	40	**0.46**
65	4	40	**0.41**	53	3	40	**0.44**
67	4	40	**0.42**	57	3	40	**0.48**
67	4	40	**0.42**	37	3	26	**0.47**
68	4	40	**0.43**	39	3	26	**0.50**
66	4	40	**0.41**	35	3	26	**0.45**
69	4	40	**0.43**	38	3	26	**0.49**

Figure 4 : Comparison of measured and simulated data

Based on the strong similarities between the real and the simulated data, we conclude that the model is an accurate approximation of vehicular mobility, in the above-mentioned, frequently-met, city scenario.

Network Simulator

The network simulator module copes with the delivery of messages from one node to another. It offers a set of network primitives that can be called by the node applications emulated on top of this simulation framework. Of special interest are the MAC and physical layers that determine VANET applications performance (Takai et.al. 2001).

At the physical layer, we use a model with cumulative noise calculation and signal reception based on SNR (Signal-To-Noise) threshold. This means that when a radio receives a signal of a given strength, the noise is calculated as the sum of all the other signals on the channel, and the ratio of the two values is the SNR. The signal can be successfully received if the value of SNR is higher than a given threshold SNRT.

The radio wave propagation can be affected by three independent phenomena: path loss, fading and shadowing (Takai et.al. 2001). The path loss effect is considered to be the most important factor and it reflects the signal power attenuation due to the propagation distance. Our simulator has two signal propagation models: free-space and plane earth two-ray path loss. While the first is an idealized model, the two-ray path loss model considers the effect of earth surface reflection and is more accurate.

Our simulator delivers a message to all the nodes in the wireless range in an optimized way using a local search of nodes. This is possible due to efficient indexing of the map points, using the PeanoKey mechanism (Dashtinezhad et. al. 2004). A PeanoKey is associated with a point in the 2D space, and it is obtained by interleaving the digits of the two coordinates. Thus, the 2D set of points is represented in a one dimensional set. For example the PeanoKey associated with the geographical point at 26.047800 degrees longitude and 44.435348 degrees latitude is 4246403457384080. When the map is being built, a set of sorted PeanoKeys is also computed, corresponding to all the points of the map. Consecutive PeanoKeys in this set correspond to points that are close on the map. In this manner the wireless environment of a node is quickly analyzed, its wireless neighbors are discovered and a map of the radio signal is built.

At the Link Layer, we have implemented the CSMA/CA channel access mechanism which is the base of IEEE 802.11 standard. The basic principles of CSMA/CA are *listen before talk* and *contention*. When a node has to send a packet, it starts by listening the environment and if idle, begins the transmission. If the medium is busy, the node waits for a random amount of time before checking again.

A lot of work has been done to study routing layer protocols in VANETs. Classical topology-based protocols have been proven to perform poorly (Blum et.al. 2004), and location-based approaches have been suggested (Festag et.al. 2004). We have implemented a geographical routing protocol for the routing layer. A node uses geographical information about its neighbors, the origin and the destination of a packet, in order to make forwarding decisions. However, due to the very high mobility and the frequent partitioning of VANETs, no guarantees can be made about reliable end-to-end communication.

Fuel Consumption and Pollutant Emissions Estimation

Estimating fuel consumption and pollutant emissions is an increasingly important matter when studying vehicular traffic. Studying the improvements which VANET applications can bring on these parameters is a possible usage of our simulator. Therefore, we have considered useful to integrate the computing of fuel consumption and pollutant emissions. The model we have implemented is influenced by the work of Akcelik and Besley (2003). Of special relevance to our work, we consider the estimation of the relation between fuel consumption and emissions and the speed and acceleration of the vehicle. The model is simplified to take into account only light vehicles. Thus, based on the motion of vehicles, our simulator's engine accurately computes the fuel consumption and pollutant emissions of each vehicle. Statistics and global measures can easily be obtained.

Traffic Scenarios Generation

We have developed a GUI which can be used to generate traffic scenarios. The user can see a graphical representation of a given TIGER map and can specify flows of vehicles. A flow consists of an entry point, an exit point, a route, and the actual vehicular flow value (in vehicles/hour/lane). The user can also specify how the flow value varies over time.

SIMULATOR EVALUATION

The integrated simulator we have developed is able to simulate around 10.000 network events per second, on a 1.6GHz uni-processor. Although this value is clearly lower than the throughput of the widely-used network simulator ns2 (which can simulate over 60.000 events per second, on a 2GHz uni-processor), it must be noted that our network simulator is integrated with a complex node mobility simulator, responsible for accurately computing the motion of all nodes, every time cycle.

As a basic vehicular computing application to experiment with our simulator, we have chosen TrafficView (Dashtinezhad et. al. 2004). This application assumes each vehicle is equipped with a GPS receiver and a wireless communication device, and has a unique identifier. Periodically, each vehicle broadcasts information about its location. This information can be forwarded further by neighboring vehicles, thus creating a platform where each vehicle is aware of its neighbors.

Our simulator is able to simulate real-time (1 second of simulation in 1 second of real time) the motion of 1000 vehicles, in a complex city-scenario: a square region of 1km by 1km, representing a part of downtown Manhattan, with a large number of intersections and traffic lights (Figure 5). While moving, all the nodes run the simple neighbor-discovery and update protocol, with a 1 second period for the beacons.

We have also successfully performed the simulation of more complex scenarios, involving up to 10.000 vehicles, and we were able to obtain significant results, in spite of the increased simulation time.

The graph in Figure 6 shows how the increase of the number of nodes influences our simulator's performance. The results are based on simulations performed in a highway traffic scenario, with all the nodes running a neighbor-discovery and update protocol, with a 1 second period for the beacon messages. The simulations were performed on a 1.6GHz uni-processor, with 512Mb of memory.

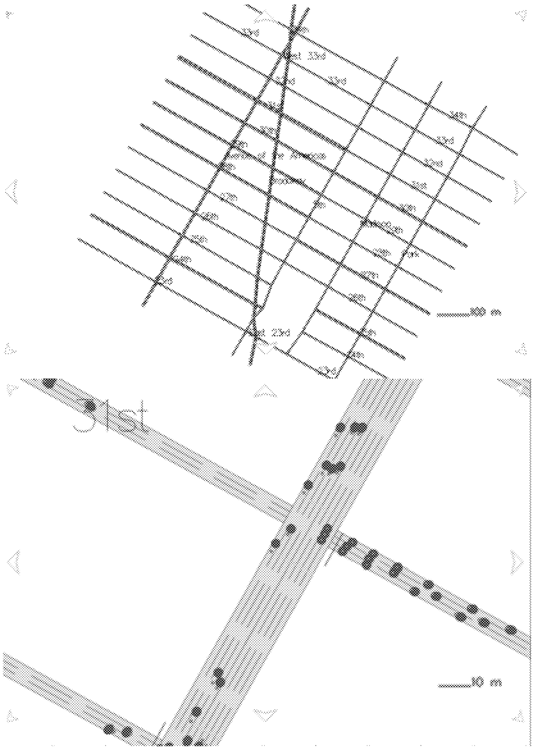


Figure 5 : Simulation screenshots

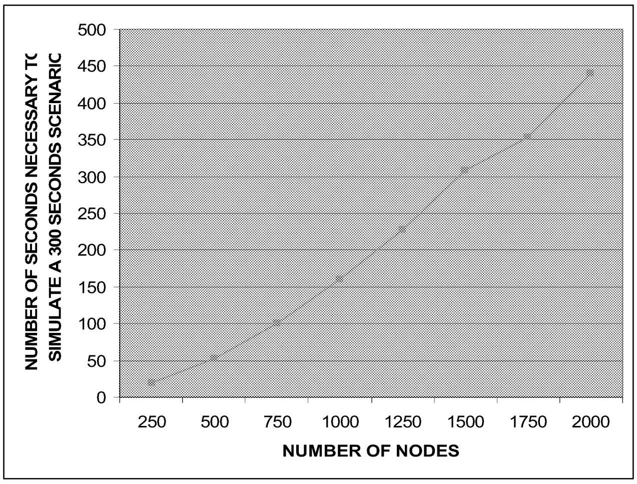


Figure 6 : Simulator performance

As previously described, three main parts can be distinguished in the simulation process: mobility, simulator engine and emulation of the nodes' application. Optionally, a graphical user interface may show simulation details. Obviously, additional time is consumed with display functions and the synchronization mechanisms. The most time consuming part of the simulation is the emulation of the application code, which should be run individually be each node (Figure 7). The TrafficView application has to parse all the incoming messages, update the local vehicle records and create new messages for broadcast. Figure 7 shows that for high densities, when the network is widely connected, messages are propagated easily from car to car and more than half of the simulation time goes on processing the messages received by each of them.

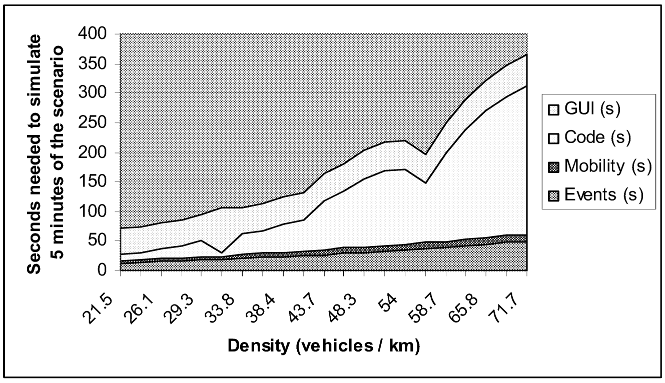


Figure 7 : Time measurements for the simulation process and its components, depending vehicles density. Test scenario: 10 km of highway with a traffic flow varying between 500 and 1500 vehicles/hour/lane.

APPLICATIONS AND FUTURE WORK

A large range of vehicular computing applications can be evaluated using our simulator. We are currently in the process of studying three such applications.

First, we are developing an adaptive traffic lights system in which wireless traffic lights can obtain real-time traffic information by communicating with vehicles. We focus the study on algorithms for efficient traffic control. Our integrated simulator allows us to easily emulate fixed nodes (traffic lights) and the application running on top of them. The vehicles move according to the traffic signals, because of the mobility model. We can easily compare traffic fluency when using different solutions for the traffic lights.

Secondly, we are studying a query-reply protocol. A node could make use of the ad-hoc network in order to obtain real-time traffic information about remote regions. This application makes use of the geographical routing protocol. We want to see in what conditions such an application can work, in the highly mobile environment of a VANET.

Finally, we are studying an application for suggesting best routes to drivers, based on real-time traffic information. By using the ad-hoc network and/or infrastructure, drivers can obtain the best route to a destination, taking into account the current traffic shape.

CONCLUSIONS

The studying of VANET protocols requires efficient, accurate simulation tools. Existing simulators which have not been designed especially for VANETs are difficult to use. We have developed an integrated simulator, comprising a complex model for vehicles mobility, a wireless network simulator and an interface for the emulation of vehicular applications. On top of this simulator we have implemented TrafficView (Dashtinezhad et.al. 2004), an application for information exchange between vehicles.

In this context, we have analyzed the performance of the simulator, and found it can be used to simulate networks of several thousands of nodes, in complex city scenarios, as well as highway scenarios. The simulator allows the evaluation of a large range of vehicular computing applications, which cannot be studied by using existing simulators.

ACKNOWLEDGEMENT

This material is based upon work supported in part by NSF under grants ANI-0121416, CNS-0520123 and CNS-0584475.

REFERENCES

Akcelik R., Besley M. 2003. “Operating cost, fuel consumption, and emission models in aaSIDRA and aaMotion”. *25th Conference of Australian Institutes of Transport Research (CAITR 2003)*

Blum J., Eskandarian A. and Hoffman L.J. 2004. “Challenges of Intervehicle AdHoc Networks”. *IEEE Transaction on Intelligent Transportation Systems*, 5(4):347-351.

Choffnes D.R. and Bustamante F.E. 2005. “An Integrated Mobility and Traffic Model for Vehicular Wireless Networks”. *Proc. of the 2nd ACM International Workshop on Vehicular Ad Hoc Networks (VANET), September 2005.*

Dashtinezhad S., Nadeem T., Dorohonceanu B., Borcea C., Kang P. and Iftode L. 2004. “TrafficView: A Driver Assistant Device for Traffic Monitoring based on Car-to-Car Communication”. *Proceedings of IEEE Semiannual Vehicular Technology Conference*

Fellendorf M. and Vortisch P. 2000. “Validation of the Microscopic Traffic Flow Model VISSIM in Different Real-World Situations”

Festag A., Fussler H., Hartenstein H., Sarma A. and Schmitz R. 2004. “FleetNet: Bringing Car-to-Car Communication into the Real World”. *Proc. Of the 11th World Congress on ITS, Nagoya, Japan.*

Mangharam R. Weller D.S., Stancil D.D., Rajkumar R., Parikh J.S. 2005. “GrooveSim: A Topography-Accurate Simulator for Geographical Routing in Vehicular Networks”. *Proc. of the 2nd ACM International Workshop on Vehicular Ad Hoc Networks (VANET), September 2005.*

Saha A. and Johnson D. 2004. “Modelling Mobility for Vehicular Ad-hoc Networks”. *ACM International Workshop on Vehicular Ad Hoc Networks (VANET)*, 2004.

Takai M., Martin J., Bagrodia R. 2001. “Effects of Wireless Physical Layer Modeling in Mobile Ad-Hoc Networks”. *ACM International Symposium on Mobile Ad Hoc Networking and Computing* (MobiHoc 2001)

US Census Bureau, Topologically Integrated Geographic Encoding and Referencing system (Tiger Line) - *http://www.census.gov/geo/www/tiger.*

Wiedemann R. 1974. “Simulation des Straßenverkehrsflusses”. Schriftenreihe des Instituts für Verkehrswesen der Universität Karlsruhe, Heft 8

Wiedemann R. 1991. “Modelling of RTI-Elements on multi-lane roads”. *Advanced Telematics in Road Transport* edited by the Comission of the European Community, DG XIII, Brussels (1991).

AUTHOR BIOGRAPHY

CRISTIAN GORGORIN, VICTOR GRADINESCU and **RALUCA DIACONESCU** obtained their Bachelor of Science in Computer Science degrees from “Politehnica” University Bucharest, in the summer of 2006. In the summer of 2005, they were invited by Rutgers University to visit the Computer Science Department, and they worked with Professor Liviu Iftode in the field of vehicular computing. This paper is partly based on work performed during that time. The paper is also based on work performed for their graduation theses, coordinated by Professor Valentin Cristea. Presently, they are all pursuing Master of Science degrees, in the Computer Science department of University “Politehnica” Bucharest.

Concept of “ hands on “ training for Spacecraft Operations.

Christian D. Bodemann, Joachim.Ochs, Carol Quirke, Roberto Palmari
Vega IT GmbH
Robert Bosch Str. 7
64392 Darmstadt
christian.bodemann@vega.de, joachim.ochs@vega.de ,
, carol.quirke@vega.de,roberto.palmari@vega.de

Abstract

Spacecraft operation is one of the most important areas in the space business. During the last decades hundreds of satellites have been launched and are operational. A lot of effort has been spent in these decades to reduce the cost of spacecraft operation. Many elements of the operation have been automated. However, operators will be always mandatory and therefore one of the key challenges in operations will remain open: how should operators be trained?

The training in itself costs time and money. Training is normally performed in two different ways:

- training on the job
- external classroom training

An operator who can perform training on the job, because he has experienced personal hasthe following costs to face:

- Time spent by experienced operators to train the inexperienced on the job,
- Time spent in preparing and organising the training,

As training to operations is not just theory, the trainee also needs to focus on practical issues, like the use of the mission control system. Therefore, one of the limitations is the availability of the hardware and software to perform this training. As space hardware and software is too expensive, normally the real mission control system or at least the redundant unit is used for this kind of training. Another limitation is when, during the programme lifecycle, to perform the
The aim is to go further than a training course, encouraging a real transfer of know-how to scientific or space agencies, universities or spacecraft operators world-wide. The Spacecraft Operations Training Centre not only provides a set of training training. Before the beginning of a mission the ground segment might be available, but not finally validated. After the launch it is nearly impossible. During the mission it becomes easier. So training will normally be performed either before the launch or after the commissioning of the satellite.

If a mission lifetime is just about five years no further training is mandatory, but should the mission lifetime be greater than five years, a natural fluctuation of operators is to be expected. This means that experienced operators with practical experience will leave the team at some point and will normally be replaced by inexperienced operators. The time and cost impact of this can not and should not be ignored.

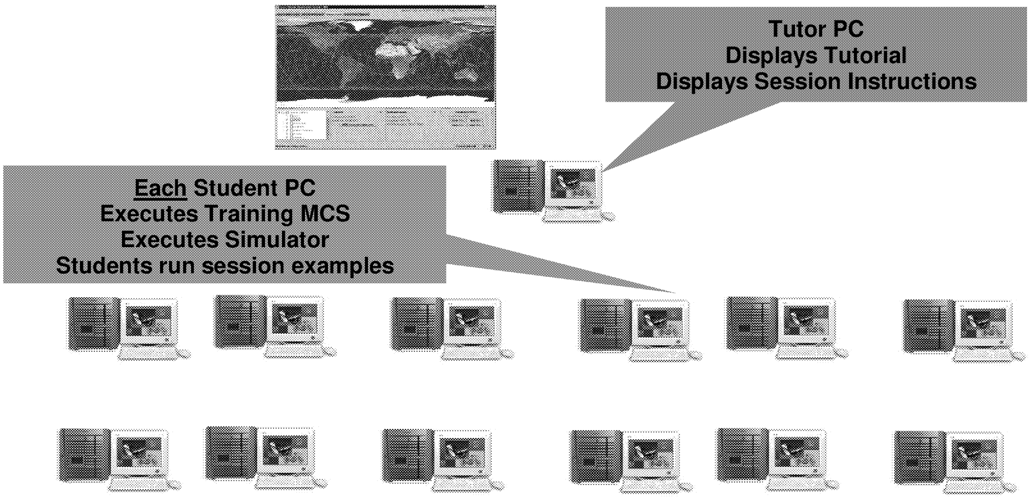
Newcomers in the space business have an even bigger problem. They usually have no experienced operators and therefore have to buy external operators–training packages. In the majority of the cases it is the satellite supplier who offers such training. This is quite expensive and very satellite specific.

The training solution: The Satellite Operations Training Centre

In the last two years VEGA has developed an approach which can help to reduce these problems, the VEGA Satellite Operations Training Centre. It provides both the facilities and the training materials to teach the fundamentals of satellite operations and related technologies. It proposes a modular approach, allowing comprehensive training to be tailored to a particular audience.
sessions, but also installs a complete training package, including instructors, hardware and software, allowing the client to teach new team members and conduct refresher training for years to come.

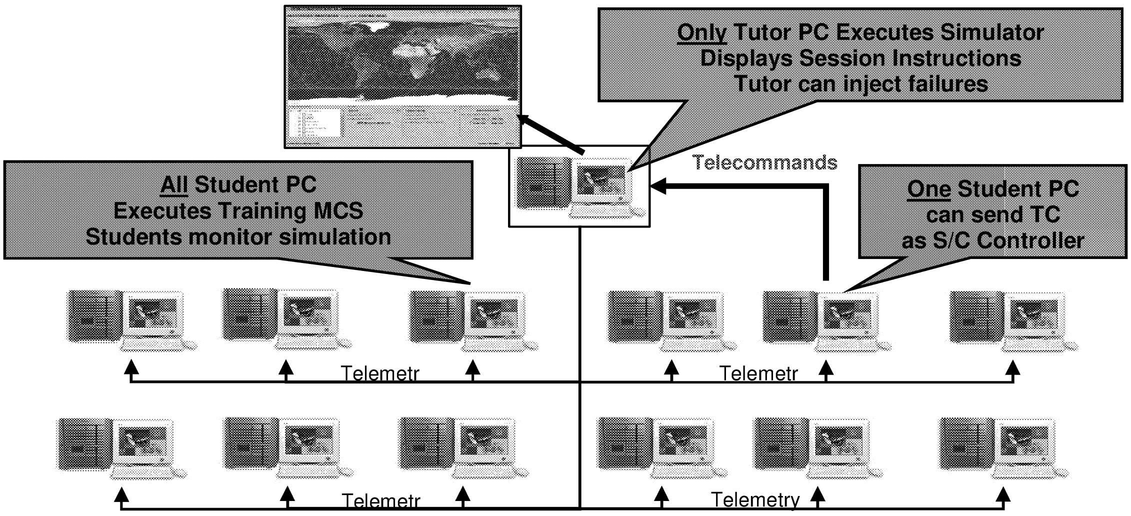
The course may be adapted to train future satellite operators or operations engineers, to provide a high level of technical know-how to space engineers or to train instructors who will take over the Mission Control System Training Facility.

The Satellite Operations Training Centre is derived from the existing software and satellite technology developed by VEGA for numerous clients including the European Space Agency, Eumetsat and EADS.



Satellite Operations Training Centre Classroom Training Configuration

The Satellite Operations Training Centre is installed in a classroom at the client's site and consists of main operations screens, a generic mission control system and a generic satellite simulator. Each trainee workstation is separately connected to the central systems on a local area network. The instructor has a workstation to drive the course material and monitor the student progress.



Satellite Operations Training Centre Team Training Configuration

Key benefits

Spacecraft Operations Training Centre offers clients the following benefits:

- ❑ Self contained training centre
- ❑ Instructor training for autonomous operation
- ❑ Hands-on training of generic satellite systems

- Detailed foundation course in satellite operations
- Refresher training for existing operators
- Ability to upgrade the system in the future to add simulations of specific satellites that the client may launch and operate
- National autonomy for satellite operations training

Target Group

The Satellite Operations Training Centre is for people who want to follow a career in spacecraft operations, starting from little or no practical experience.

It is designed for organisations willing to train staff to become Spacecraft Controllers or Satellite Operations Engineers.

After the participation of the Satellite Operations Training Centre the trainee:

- knows what equipment and components are needed to operate a satellite
- has learned the physical background to understand how a satellite is operated
- knows standard procedures to operate a satellite
- has operated a simulated training satellite and gained in this way first experience in satellite operation

Initial Qualification

The initial qualifications needed to benefit from the Spacecraft Operations Training Centre are:

University bachelors level qualification in an engineering discipline such as mechanical or electrical engineering. Physics can be also considered. People with relevant or similar professional experience.

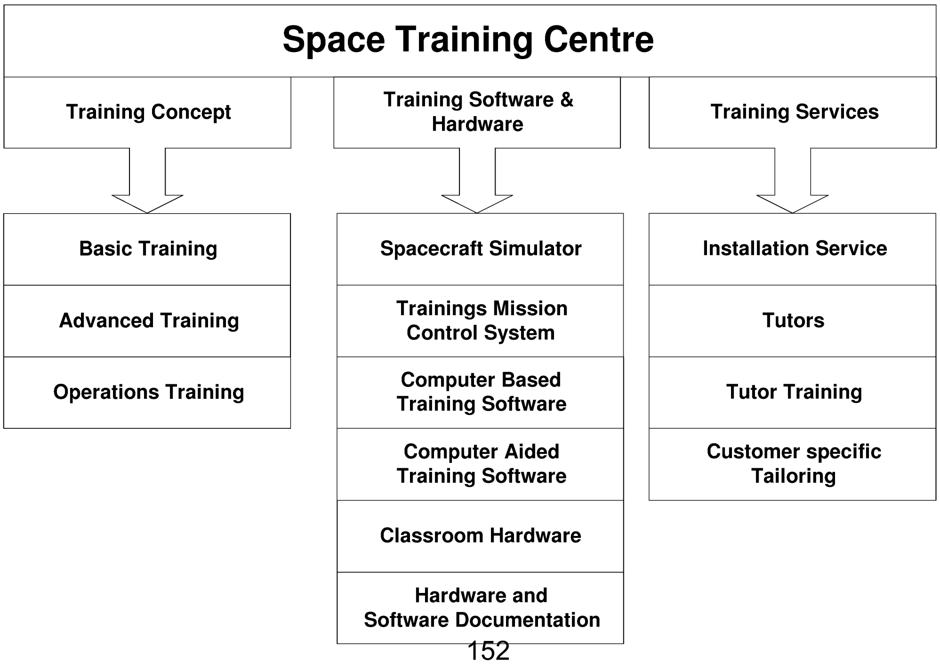
The Training Course

This section provides details on the Satellite Operations Training Centre training solution.
The Spacecraft Operations Training Centre is a component-based product, which includes a training concept, training software and hardware and training services.

Spacecraft Operatiopns Training Centre Structure

The Spacecraft Operations Training Centre foresees three training levels:

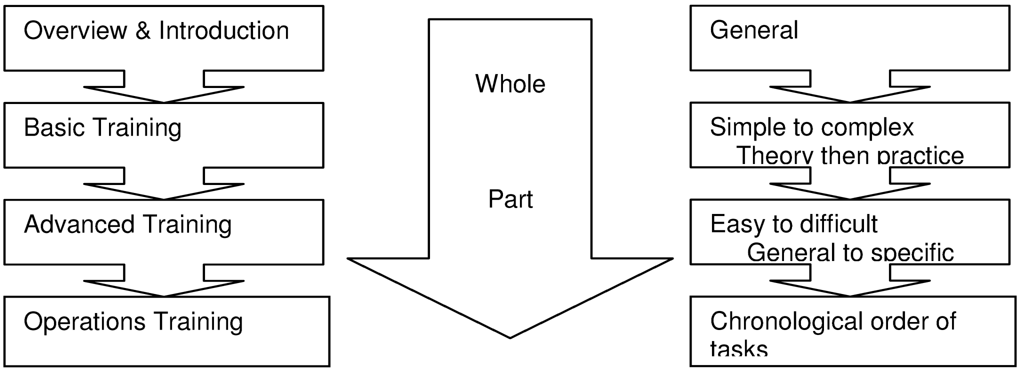
- Basic Training: Goal of the Basic Training is to harmonise the level of knowledge. At the end of this part there is a common level of knowledge. The foreseen duration is one week.
- Advanced Training: Goal of the Advanced Training is to enable the trainee to command and monitor every


Space Training Centre
Training Concept
Training Software & Hardware
Training Services
Basic Training
Advanced Training
Operations Training
Spacecraft Simulator
Trainings Mission Control System
Computer Based Training Software
Computer Aided Training Software
Classroom Hardware
Hardware and Software Documentation
Installation Service
Tutors
Tutor Training
Customer specific Tailoring

spacecraft subsystem. This part includes all parts of the familiarisation with the Spacecraft Operations Engineer job. The foreseen duration is four weeks.

- Operations Training: Goal of the Operations Training is to enable the trainee to operate a spacecraft in simple and nominal cases. In this part simple LEOP session should be considered. As a final exam a small mission scenario could be foreseen. The foreseen duration is one week.

Training Strategy Overview



Spacecraft Operatiopns Training Centre Training Strategy

Training Centre Components

The Satellite Operations Training Centre consists of the following components:

- Training Satellite Simulator including the run-time Environment
- Training Mission Control System
- Computer aided training material (Power Point Presentations)
- Classroom and classroom infrastructure
- Documentation

In addition, the trainer provides a formal report for all the trainees attending the course detailing their performance during the training courses.

Satellite Simulator

The Spacecraft Operations Training Centre includes a satellite simulator, which is entirely software based and runs on a standalone PC. On each of the PC's the training mission control system and the simulator are installed so that each trainee can individually perform exercises during the different training sessions. During the operations training all trainees' PC are connected to one simulator running on the tutor PC via TCP/IP. The simulator is generic and specifically configured and set up for training purposes. The Simulator includes all typical subsystems and instruments a satellite in a low earth orbit is using. Therefore all operations are similar or identical to real equipment.

Conclusion

Spacecraft Operations Training is quite a challenging problem. Operators of satellites need trained operations engineers. The required training is usually done on the job, so during the mission lifecycle. This represents a considerable cost, just consider the delay needed to bring a non experienced engineer to the required knowledge level. Our paper is aiming to show that this training can be performed with a “hands on” approach and outside the scope of a particular mission. This training can be performed inside universities or at spacecraft operator premises in order to qualify Spacecraft Operations engineers. The Spacecraft Operations Training Centre has already been delivered so far to one university.

FLUID FLOW SIMULATION

INVESTIGATION OF FLOW DYNAMICS IN POROUS MEDIA USING COMPUTER SIMULATION

Arezou Jafari

Department of Energy and Environmental
Engineering, Lappeenranta University of
Technology, 53850, Lappeenranta, Finland

Piroz Zamankhan

Laboratory of Computational Fluid and Biofluid
Dynamics, Lappeenranta University of
Technology, 53850, Lappeenranta, Finland

S. Mohammad Mousavi
Department of Chemical and Petroleum
Engineering, Sharif University of Technology,
Tehran, Iran
Department of Chemical Engineering,
Lappeenranta University of Technology, 53850,
Lappeenranta, Finland

Kari Pietarinen

Department of Energy and Environmental
Engineering, Lappeenranta University of
Technology, 53850, Lappeenranta, Finland

Pertti Sarkomaa
Department of Energy and Environmental Engineering, Lappeenranta University of Technology,
53850, Lappeenranta, Finland

KEYWORDS
Porous, Simulation, Flow Dynamics, Turbulence, Large Eddy Simulation, Reynolds Stress Model.

ABSTRACT

The aim of this work is to investigate flow hydrodynamics in porous media. Random non-overlapping spherical particles in a cylindrical geometry were produced as a porous media. Navier-Stokes equations in three-dimensional porous packed bed have been solved and dimensionless pressure drop has been studied for a fluid flow through the porous media at different Reynolds numbers (based on pore permeability and interstitial fluid velocity). The numerical results are in good agreement with those reported by Macdonald (1979) in the range of Reynolds numbers studied. At higher Reynolds numbers turbulent models such as large eddy simulation (LES) and Reynolds stress model (RSM) also have been employed for modeling turbulence flows through irregular array of particles. The results obtained show that LES compare to RSM can predict the fluid flow better in high Reynolds numbers. Also it is found that in turbulent regime, laminar model can predict the flow dynamics as well as LES, and it is clear that working with laminar model is easier than turbulent models.

INTRODUCTION

Fluid flow in a granular bed, as illustrated in Fig. 1, has attracted attention due to its importance in many industrial processes, such as chromatographic separation technology (Lightfood et al. 1981), packed bed reactor and contacting device design (Strigle 1994), modeling of contaminant transport in hydro-geological and environmental systems (Slichter 1905), and studies of perfusion in biological media (Kostyuk and Krishtal 1984).

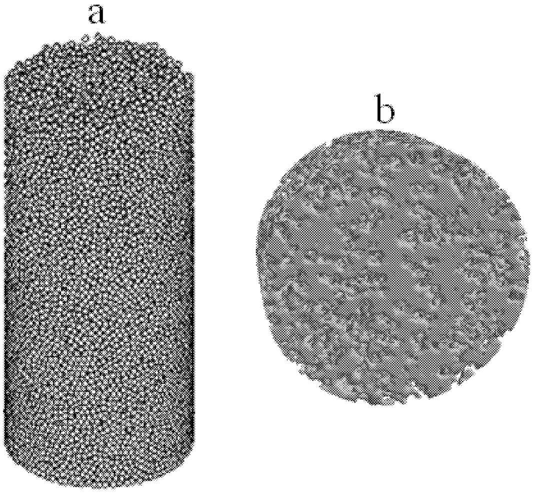


Figure 1: (a) Illustration of particles for the simulation in this study. (b) A three dimensional view of the void region within an irregular array as shown in (a).

According to Fand et al. (1987), four different regimes through porous media were identified as pre-Darcy, Darcy, Forchheimer, and turbulent. Henry Darcy (1856) observed that under certain conditions the volume rate of water through a pipe packed with sand was proportional to the negative of the pressure gradient. This relationship is known as Darcy's law. Darcy flow is an expression of the dominance of viscous forces applied by the solid porous matrix on the interstitial fluid and is of limited applicability. Post-Darcy regimes are affected by inertia forces and turbulence. Forchheimer (1901) first suggested a non-linear relationship between the pressure gradient and the fluid velocity. At high Reynolds numbers, the nonlinear contribution from the convective terms in the Navier-Stokes equations becomes relevant. The inertial effect on the velocity field is consistent with the presence of several vortices and zones of flow separation (Andrade et al. 1997).

A key question is under what conditions flow in porous media could become turbulent. The fundamental understanding of the transition from laminar to turbulent convection in porous media is far from being conclusive. While major efforts are under way, there is still a significant challenge in front of the scientists and engineers to uncover the complex behavior linked to this transition.

Nield (2001) reviewed published papers involving models of turbulence in porous media that discussed matters such as inertial effects, lateral momentum transfer and spin-up, nonlinear drag and the detection of the onset of turbulence. Considering his arguments, a significant challenge faces researchers to explore the complex behavior linked to the detection of the onset of turbulence in a porous medium.

LES can be used to calculate flow statistics, which are determined by the larger scales, such as the mean velocity and second-order velocity moments. Indeed, these quantities are often required in practice. Recent advances in physical models, numerical techniques, and computational power together have made LES (Ghosal and Moin 1995) a useful tool for computing gas-particle flows in vertical channels (Yamamoto et al. 2001) where the dynamics of larger scales is influenced by the presence of small scales because of nonlinear interactions.

The objective of the current work is to investigate fluid flow through a bed of non-overlapping spherical particles in a cylindrical geometry as porous medium. The Navier-Stokes equations were solved for the velocity and pressure fields in the fluid phase of the pore space by discretization using the control volume method (Patankar 1980). At high Reynolds numbers different turbulent models such as LES and RSM beside laminar model were investigated, and obtained results have been compared with previous works (Ergun (1952), Macdonald et al. (1979), Fand et al. (1987), and Kececioglu and Jiang (1994)).

MATHEMATICAL FORMULA

Figure 1(a) illustrates non-overlapping uniform size spheres randomly distributed within a cylinder. The method described in (Zamankhan et al. 1999) was used to generate the random media. The spherical particles as well as the cylinders are impermeable to the continuous phase, namely the fluid. Different numbers and diameters of particles were tested to obtain different porosity. Also longer cylinders were applied to show that porosity has more effect on dimensionless pressure drop compare to length of the tubes.

The flow is assumed to be horizontal, steady state, incompressible and isothermal. The mathematical description for the flow of a viscous fluid through three dimensional granular bed is based on the steady form of the Navier-Stokes and continuity equations (Schlichting 1979) for momentum and mass conservation, respectively. The equations of motion may be written in the following form:

$$\begin{aligned} &\rho u.\nabla u = -\nabla p + \mu \nabla \cdot \nabla u, \\ &\nabla \cdot u = 0. \end{aligned} \qquad (1)$$

where $\rho, p, \mu, and\ u$ are density, dynamic viscosity, pressure and velocity of the fluid respectively.

A uniform velocity profile was assumed at the inlet whereas the pressure at the exit is assumed to be fixed to the local atmospheric pressure. In addition, no-slip boundary condition at the entire solid fluid interface is considered. Equations (1) were solved numerically for the pressure and velocity fields using the finite volume method with the pressure correction algorithm SIMPLE (Patankar 1980). In this work, dimensionless pressure drop has been studied for a fluid flow through the porous media at different Reynolds numbers. It would be expected that by increasing the fluid inertial forces the transition from laminar to turbulent regime should be observed. There exist some experimental reports such as Mickeley et al. (1965), Kirkham (1968) and Dybbs and Edwards (1984) that proved the existence of a turbulent within a saturated porous media. So in this paper at higher Reynolds numbers both laminar and turbulence models were considered. Performing Direct Numerical Simulation (DNS) in which all scales of the flow are properly resolved for simulating flows in models such as that illustrated in Fig. 1 is not currently feasible due to prohibitive computational requirements. Here turbulent models such as LES and RSM have been studied.

In the following, the application of LES (Sagaut 2002) which has the less ambitious goal of describing the larger scales of the flow field through a stationary irregular array of particles was investigated. Using LES the dynamic range of scales to be resolved was reduced by filtering operation performed on the Navier-Stokes equations, so LES generates an approximation in which scales below the filter size are missing. The turbulent energy cascade generates smaller scales and all scales of turbulence are dynamically significant. Given the lack of small scales below a certain size, the correction must be applied via the aforementioned additional terms (known as subgrid stress tensor) in the governing equations of LES. The subgrid scale (sgs) stress tensor describes the effect of the unresolved scales on the larger resolved scales. The replacement of sgs stress by an explicit physical model is required to close equations for the large scale fields on a grid small enough (but much larger than the Kolmogorov scale) to provide reasonable resolutions. Details of the LES equations used in the present study are given in below.

Filtered Navier-Stokes equations for an incompressible viscous Newtonian fluid may be obtained applying a spatial filter such as

$$G\left(x_i-\xi_i\right)=\left(\bar{\gamma}/\pi\bar{\Delta}^2\right)^{1/2}\exp\left(-\bar{\gamma}\left|x_i-\xi_i\right|^2/\bar{\Delta}^2\right) \qquad (2)$$

where $\bar{\gamma}$ is a constant whose value is usually chosen to be 6, $\bar{\Delta}$ represents cutoff length, and G shows spatial filter. The filtered momentum equation may be given as

$$\frac{\partial\bar{u}_i}{\partial t}+\frac{\partial}{\partial x_j}\left(\overline{u_iu_j}\right)=-\frac{\partial\bar{p}}{\partial x_i}+\nu\frac{\partial}{\partial x_j}\left(\frac{\partial\bar{u}_i}{\partial x_j}+\frac{\partial\bar{u}_j}{\partial x_i}\right), \qquad (3)$$

where ν represents the kinematic viscosity, and $\bar{p}$ represents the filtered pressure. Note that the resolved part of a space-time variable $\phi\left(x_i,t\right)$ is defined as

$$\bar{\phi}\left(x_i,t\right)=\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\phi\left(\xi_i,t'\right)G\left(x_i-\xi_i,t-t'\right)dt'd\xi_1d\xi_2d\xi_3.$$

(4)

The filtered momentum equation includes a non-linear term, $\overline{u_iu_j}$, which have to be expressed as a function of $\bar{u}_i$ and u_i' defined as u_i-u_i'. The filtered continuity equation is given as

$$\partial\bar{u}_i/\partial x_i=0. \qquad (5)$$

Using Leonard's decomposition (Leonard 1974) the non-linear term in Eq. (3) may be given as

$$\overline{u_iu_j}=\overline{\bar{u}_i\bar{u}_j}+\overline{\bar{u}_iu_j'}+\overline{\bar{u}_ju_i'}+\overline{u_i'u_j'}. \qquad (6)$$

Substituting Eq. (6) into Eq. (3), the filtered momentum equation may be expressed as

$$\frac{\partial\bar{u}_i}{\partial t}+\frac{\partial}{\partial x_j}\left(\overline{\bar{u}_i\bar{u}_j}\right)=-\frac{\partial\bar{p}}{\partial x_i}+\nu\frac{\partial}{\partial x_j}\left(\frac{\partial\bar{u}_i}{\partial x_j}+\frac{\partial\bar{u}_j}{\partial x_i}\right)-\frac{\partial\bar{\tau}_{ij}}{\partial x_j} \qquad (7)$$

where $\bar{\tau}_{ij}$ is called the subgrid tensor, defined as $\bar{\tau}_{ij}=C_{ij}+R_{ij}=\overline{u_iu_j}-\overline{\bar{u}_i\bar{u}_j}$. Here, $C_{ij}=\overline{\bar{u}_iu_j'}+\overline{\bar{u}_ju_i'}$ represents the interaction between large and small scales, and $R_{ij}=\overline{u_i'u_j'}$ represents the interaction between the subgrid scales.

Note that the $\overline{\bar{u}_i\bar{u}_j}$ term in Eq. (7) requires a second application of the filter. To remedy this, further decomposition as follow is required

$$\overline{\bar{u}_i\bar{u}_j}=\left(\overline{\bar{u}_i\bar{u}_j}-\bar{u}_i\bar{u}_j\right)+\bar{u}_i\bar{u}_j=L_{ij}+\bar{u}_i\bar{u}_j, \qquad (8)$$

where L_{ij} represents interactions among the large scales. Using Eq. (8), the filtered momentum equation can be given as

$$\frac{\partial\bar{u}_i}{\partial t}+\frac{\partial}{\partial x_j}\left(\bar{u}_i\bar{u}_j\right)=-\frac{\partial\bar{p}}{\partial x_i}+\nu\frac{\partial}{\partial x_j}\left(\frac{\partial\bar{u}_i}{\partial x_j}+\frac{\partial\bar{u}_j}{\partial x_i}\right)-\frac{\partial\tau_{ij}}{\partial x_j}, \qquad (9)$$

where $\tau_{ij}=L_{ij}+C_{ij}+R_{ij}=\overline{u_iu_j}-\bar{u}_i\bar{u}_j$. In order to model the subgrid tensor τ_{ij}, Smagorinsky (1963) postulated that $\tau_{ij}=-2\nu_{sgs}\bar{S}_{ij}+1/3\,\bar{\tau}_{kk}\delta_{ij}$ with $\nu_{sgs}=\left(C_s\bar{\Delta}\right)^2\left(2\bar{S}_{ij}\bar{S}_{ij}\right)^{1/2}$. The aforementioned expression is akin to a mixing-length formula with mixing-length $C_s\bar{\Delta}$. The resolved strain rate is defined as $\bar{S}_{ij}=1/2\left(\partial\bar{u}_i/\partial x_j+\partial\bar{u}_j/\partial x_i\right)$ and the constant C_s for a case of shear flows is given as $C_s\approx0.1$ (Meneveau 1994). This coefficient must decrease for flows near solid boundaries by using the damping function given as $C_{sb}=C_s^2\left[1-\exp\left(-\left(y^+/A^+\right)^3\right)\right]$, where $y^+=yu_\tau/\nu$ characterizes distances from the wall in viscous wall units, u_τ and ν are the friction velocity, and kinematic viscosity of the fluid, respectively. Here, A^+ represents a constant whose value is set to 26.

In this study beside laminar and LES at high Reynolds numbers RSM method was also investigated. Assuming that the flow is at a high Reynolds number, the governing equations for the conservation of mass and momentum are averaged over both time and space (Raupach 1982), and $\overline{u'_i\cdot u'_j}$ has been calculated using differential transport equations. Various sub-models proposed by Speziale et al. (1991), Mellor and Herring (1973), and Rotta (1951) were used for the pressure–strain correlation term, the turbulent diffusion term, and the dissipation term, respectively. The choice of these models is based on the numerical experiments where details of these sub-models can be found in (Choi and Kang 2001). The complete and general model equations for the Reynolds stress (R_{ij}) are given as

$$\begin{aligned}\bar{u}_k\frac{\partial R_{ij}}{\partial x_k}&=-\left(R_{ik}\frac{\partial\bar{u}_j}{\partial x_k}+R_{jk}\frac{\partial\bar{u}_i}{\partial x_k}\right)-\frac{2}{3}\varepsilon\delta_{ij}\\&+c_s\frac{\partial}{\partial x_k}\left[\frac{k^2}{\varepsilon}\left(\frac{\partial R_{ij}}{\partial x_k}+\frac{\partial R_{ik}}{\partial x_j}+\frac{\partial R_{jk}}{\partial x_i}\right)\right]\\&+\alpha_0\varepsilon b_{ij}+\alpha_1\varepsilon\left(b_{ik}b_{jk}-\frac{1}{3}b_{mn}b_{nm}\delta_{ij}\right)+\alpha_2ks_{ij} \qquad (10)\\&+\alpha_3p_kb_{ij}+\alpha_4k\left(b_{ik}s_{jk}+b_{jk}s_{ik}-\frac{2}{3}b_{kl}s_{kl}\delta_{ij}\right)\\&+\alpha_5k\left(b_{ik}W_{jk}+b_{jk}W_{ik}\right)\end{aligned}$$

where the first term in the right-hand side of the above equation is the production term, the second term is the dissipation rate, the third term is the turbulent diffusion, and the other terms are the pressure–strain correlations. In Eq.

(10), c_s is a model coefficient (=0.22/3), k and ε are the turbulent kinetic energy and its dissipation rate, respectively, p_k is the production of turbulent kinetic energy, b_{ij} is the anisotropy tensor, δ_{ij} is the Kronecker delta, s_{ij} is the rate of strain tensor, W_{ij} is the rotation tensor, and $\alpha_0 - \alpha_5$ are six empirical coefficients. For these parameters the following values have been used $\alpha_0 = -3.4, \alpha_1 = 4.2, \alpha_2 = 0.8 - 1.3(b_{mn}b_{nm})^{0.5}, \alpha_3 = -1.8,$ $\alpha_4 = 1.25, and\ \alpha_5 = 0.4.$

In Eq. (10) the dissipation rate of k can be obtained by solving the standard ε-transport equation such as

$$\bar{u}_j \frac{\partial \varepsilon}{\partial x_j} = \frac{\partial}{\partial x_k}\left(C_\varepsilon \frac{k}{\varepsilon} R_{kl} \frac{\partial \varepsilon}{\partial x_l} \right) + \frac{\varepsilon}{k} C_{\varepsilon 1} p_k - C_{\varepsilon 2} \frac{\varepsilon^2}{k} \qquad (11)$$

where $C_\varepsilon (= 0.18), C_{\varepsilon 1} (= 1.45), and\ C_{\varepsilon 2} (= 1.9)$ are empirical constants.

RESULTS AND DISCUSSION

Figure 2 illustrates the plots of dimensionless pressure drop, $\left(-dp/dz\right) K / \mu V_f\ \phi$, versus the Reynolds number, $\mathrm{Re}_K = \rho V_f \sqrt{K/\phi} / \mu$, based on pore permeability, K, and interstitial fluid velocity, V_f, as suggested by Kececioglu and Jiang (1994). Here, $\phi, and\ -\frac{dp}{dz}$ represent porosity and pressure drop, respectively. A Darcy regime can be observed for numerical results over a range of Reynolds numbers for which the dimensionless pressure drop is equal to a constant. From this part of the curves, the porous medium permeability was determined. Two post-Darcy regimes have been shown and the change in the slope in Fig. 2 (a) and (b) indicates the transition from Forchheimer regime to the turbulent flow. It is clear from comparing Fig. 2 (a) and (b) that the flow regime demarcation varies with permeability. In addition, Kececioglu and Jiang (1994) from their experimental work showed that particle diameter has effect on flow regime. Note that the transition criteria from laminar to turbulent flow for flow through porous media have not been yet defined**.** A comparison between the numerical results and the correlations proposed by other researchers mentioned in table 1, implies that the results are in good agreement with Macdonald's results (Macdonald 1979) in the actual flow Reynolds numbers studied here. Slightly there is a difference in slopes when transition to the second post Darcy regime occurs, but the maximum error is less than 10 percent.

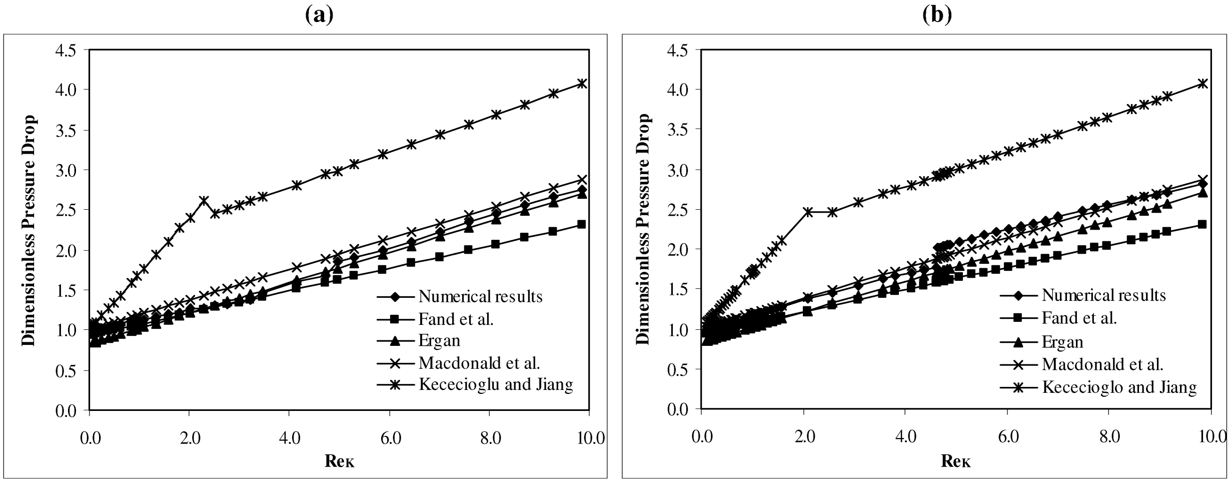


Figure 2: Plot of dimensionless pressure drop versus Re_K for systems with two different porosities: (a) porosity=0.5, and (b) porosity=0.7. Here length and diameter of the cylinder are 21 and 6 cm respectively.

In table 1 finding of Ergun (1952), Macdonald et al. (1979), Fand et al.(1987), and Kececioglu and Jiang (1994) were listed. The transition from laminar to turbulent flow in a granular bed seems rather subtle, compared to the rather sharp transition that occurs in the pipe flow. The occurrence of relaminarization from turbulent to laminar regime is quite likely leading to the conditions at which combination of different flow types exists. The existing literature provides limited quantitative information on the criteria that can be employed for determining the flow regime.

Table 1: Correlations for dimensionless pressure drop versus Reynolds number for flow through porous media (Kececioglu and Jiang 1994)

	Forchheimer flow	Turbulent flow
Ergun (1952)	$\frac{P'K}{\mu v} = 0.83 + 0.19\,\hat{\mathrm{R}}\,e_K\,; 0.08 < \hat{\mathrm{R}}\,e_K < 196$	
Macdonald et al. (1979)	$\frac{P'K}{\mu v} = 1 + 0.19\,\hat{\mathrm{R}}\,e_K\,;\ 0.003 < \hat{\mathrm{R}}\,e_K < 32.7$	
Fand et al. (1987)	$\frac{P'K}{\mu v} = 0.93 + 0.14\,\hat{\mathrm{R}}\,e_K\,;$ $0.57(\pm 0.06) < \hat{\mathrm{R}}\,e_K < 9(\pm 0.6)$	$\frac{P'K}{\mu v} = 1.14 + 0.12\,\hat{\mathrm{R}}\,e_K\,;$ $\hat{\mathrm{R}}\,e_K > 13.5$
Kececioglu and Jiang (1994)	$\frac{P'K}{\mu v} = 1(\pm 0.15) + 0.7(\pm 0.15)\,\hat{\mathrm{R}}\,e_K$	$\frac{P'K}{\mu v} = 1.9(\pm 0.1) + 0.22(\pm 0.04)\,\hat{\mathrm{R}}\,e_K$

In addition to laminar model for higher Reynolds numbers, different turbulent models have been studied. Table 2 represents petty of the results, which compares LES and RSM models. Numerical results show that LES model with using Smagorinsky for subgrid scale model have a better agreement with results of other researchers specially Macdonald et al. (1979). Indeed turbulence in the mentioned system is a controversial issue. Three-dimensional fluctuations occur on length scales that range from very small pores with the size of fraction of particle diameter to the scales much larger than particle diameter and on a correspondingly broad range of time scales. Hence, it is necessary to describe fluid flow in a wide range of length and time scales. Modeling of this system requires a mathematically rigorous modeling methodology capable of predicting coupling behaviors from the very small scales through full-scale system.

Table 2: Effect of LES and RSM models on dimensionless pressure drop at constant Reynolds number and porosity

Numerical results	Macdonald et al. (1979)	Fand et al. (1987)	Ergun (1952)	Kececioglu and Jiang (1994)
1.929 (LES)	1.916	1.605	1.746	2.961
2.272 (RSM)	1.916	1.605	1.746	2.961

The results appear to be grid independent. A grid independency check has been conducted to ensure that the results from the runs are not grid dependent. To do this test, three different grids have been chosen. Their details and numerical results with using LES model are shown in table 3. There was no significant variation in the dimensionless pressure drop obtained on the grid with 2×10^6 elements and those obtained from the fine grid. So the grid with 2×10^6 elements was used for all calculations.

Table 3: Effect of mesh on dimensionless pressure drop in this study

Grid	1	2	3
Numerical results (dimensionless pressure drop)	1.56	1.929	1.920
Number of tetrahedral elements	5×10^5	2×10^6	6×10^6
Number of nodes	$\approx 2\times10^5$	4×10^5	9×10^5

Fluid flow in porous media is complex especially in high Reynolds numbers. Term of turbulent flow suggested in the literature (for example see (Scheidegger 1960)) is for second post Darcy regime. The critical Reynolds number at which the transition takes place in pipes is several orders of magnitude higher than that of flow through porous media (Bear 1988). Attempts to explain these discrepancies by the heterogeneity of the medium (for example, actual turbulence starts at the larger pores while in smaller ones flow is still laminar) also fail when they are analyzed thoroughly (Scheidegger 1960).

Two types of vortices, the void and pseudo vortices play an important role in transport mechanism of turbulent flow through porous media (vafai 2000). Forced flow distortion due to the interruption of the solid particles will transport the fluid lump far away and cause associated exchange of momenta. This momentum diffusion is called as the mixing structures can play an important role in the transport mechanism such as momentum dispersion due to the solid obstruction. Fluid layers close to the walls tend to move faster resulting in a flattened velocity profile. In porous medium relaminarization could occur after diverging sections as well as turbulence enhancement after converging sections. In this light, LES is a potentially powerful tool for providing detailed and accurate solution of flow in a porous medium. In order to visualize the fluid behavior, the fluid streamlines as shown in Fig. 3 (b) were color coded using velocity magnitude.

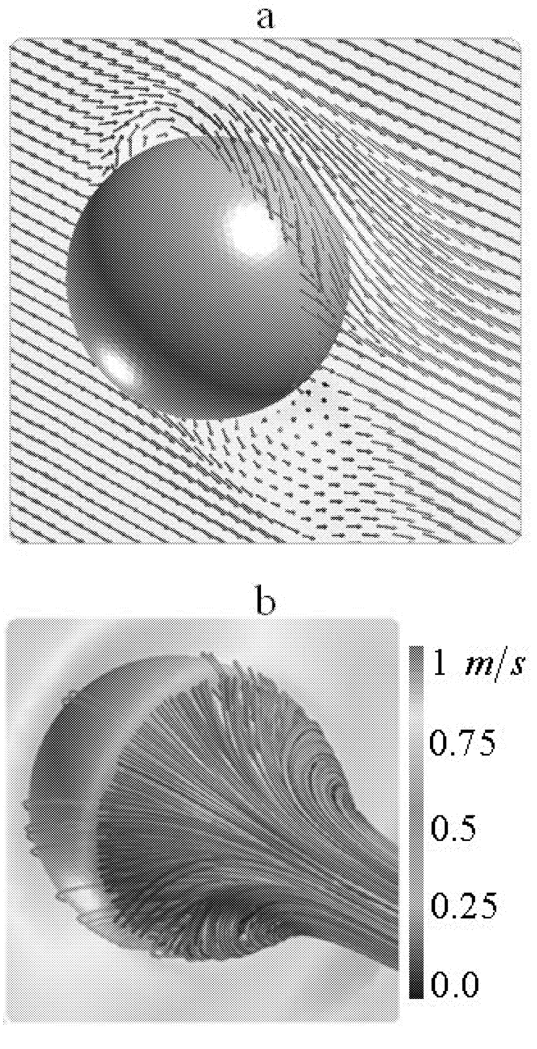


Figure 3: (a) Velocity vector field around non-overlapping spheres in the porous media. (b) Fluid streamlines passed spherical solid particles. Vectors and streamlines are color coded by velocity magnitude.

CONCLUSIONS

Numerical study of flow through random packing of non-overlapping spheres in a cylindrical geometry was investigated. Firstly, simulations were done using a model based on the Navier-Stokes equations, including inertial terms but without a turbulence model, for range of conditions studied in the second post-Darcy ("turbulent") flow regime to examine the fluid flow in a granular bed. Simulation results for pressure drop across the bed agreed well with the correlation of Macdonald (1979) for the range of actual flow Reynolds studied in this paper. LES and RSM have been used as turbulent models, and comparing the results showed that LES could forecast better than RSM. In addition, it was found that laminar model could predict the flow through porous media as well as LES.

REFERENCES

Andrade, J.S.; M. P. Almeida; J. Mendes Filho; S. Havlin; B. Suki; and H.E. Stanley. 1997. "Fluid Flow through porous Media: The Role of Stagnant Zones.", *Physical Review Letters* 79, No.20, 3901-3904.

Bear J. 1988. "*Dynamics of Fluids in Porous Media*". Dover Publisher, Inc., Mineola, New York.

Choi, S.U. and H. Kang. 2001. "Numerical Tests of Reynolds Stress Models in the Computations of Open-Channel Flows." In *Proceedings of 8th flow modeling and turbulence measurements*, Tokyo, Japan, 71-78.

Darcy H. 1856. "*Les Fontaines Publiques de la Ville de Dijon*". Ed.Victor Dolmont , Paris, France.

Dybbs, A. and R.V. Edwards. 1984. "A New Look at Porous Media Fluid Mechanics-Darcy to Turbulent". In "*Fundamentals of Transport Phenomena in Porous Media*", Bear, J. and M.Y. Corapcioglu (Eds.). Martinus Nijhoff Publishers, 199-254.

Ergun S. 1952. "Fluid Flow through Packed Columns." *Chem. Eng. Prog.* 48, 89 – 94.

Fand, R. M; B.Y. Kim; A. C. C. Lam, and R.T. Phan. 1987. "Resistance to the Flow of Fluids through Simple and Complex Porous Media Whose Matrices Are Composed of Randomly Packed Spheres." *J. Fluids Engineering* 109, 268 – 273.

Forchheimer P. 1901. “Wasserbewegung durch Boden.” *Zeischrift. Verein Deutscher Ingenieure* 45, 1782-1788.

Ghosal, S. and P. Moin. 1995. “The Basic Equations of the Large Eddy Simulation of Turbulent Flows in Complex Geometry.” *Comput, J. Phys*. 118, 24-37.

Kececioglu, I. and Y. Jiang. 1994. “Flow Through Porous Media of Packed Spheres Saturated with Water.”, *J. Fluid Engineering* 116, 164-170.

Kirkham C. E. 1968. “Fundamental Flows in Porous Media.” *Bull. Int. Ass. Sci. Hydrol.* 13, No. 2, 126-141.

Kostyuk, D.G. and O.A. Krishtal. 1984. “*Intracellular Perfusion of Excitable Cells*”. Wiley, New York.

Leonard A. 1974. “Energy Cascade in Large Eddy Simulations of Turbulent Fluid Flows.” *Adv. Geophys.* 18A, 237-248.

Lightfood, E.N.; A.S. Chiang; and P.T. Noble. 1981. “Field-Flow Fractionation (Polarization Chromatography).” *Ann. Rev. Fluid Mech*. 13, 351-378.

Macdonald, I.F.; M.S. EI-Sayed; and A.L. Dullien. 1979. “Flow through Porous Media–The Ergun Equation Revisited.” Ind. *Eng. Chem. Fundam*. 18, No. 3, 199- 207.

Mellor, G.L. and H.J. Herring. 1973. “A Survey of Mean Turbulent Field Closure.” *AIAA J.* 11,590–599.

Meneveau C. 1994. “Statistics of Turbulence Subgrid-Scale Stresses: Necessary Conditions and Experimental Tests.” *Phys. Fluids A* 6, No. 2, 815-833.

Mickeley, H.S.; K.A. Smith; and E.I. Koechak. 1965. “Fluid Flow in Packed Beds.” *Chemical Engineering Science* 23, 237-246.

Nield D.A. 2001. “Alternative Models of Turbulence in a Porous Medium, and Related Matters.” *J. Fluid Engineering* 123, 928-931.

Patankar S.V. 1980. “Numerical Heat Transfer and Fluid Flow”. Hemisphere, Washington, D.C.

Raupach M.R. and R.H. Shaw. 1982. “Averaging Procedures for Flow within Vegetation Canopies.” *Boundary Layer Meteorol* 22, 79–90.

Rotta J.C. 1951. “Statistiche Theorie Nichthomogener Turbulenz.” *Zeitschrift fur Phys* 129, 547–572.

Sagaut P. 2002. “*Large Eddy Simulation for Incompressible Flows*”. second ed., Springer, Berlin.

Scheidegger A.E. 1960. “*The Physics of Flow Through Porous Media*”. 2nd ed., University of Toronto Press, Toronto.

Schlichting H. 1979. “Boundary-Layer Theory”. McGraw-Hill, New York.

Slichter C.S. 1905. “Field Measurements of the Rate of Movement of Underground Water.” *Water Supply Papers* 14R, US. Geological Survey.

Smagorinsky J. 1963. “General Circulation Experiments with the Primitive Equations, Part I, The Basic Experiment.” *Monthly Weather Rev*. 91, 99-164.

Speziale C.G.; S. Sarkar; and T. Gatski. 1991. “Modeling the Pressure Strain Correlation of Turbulence: An Invariant Dynamical Systems Approach.” *J. Fluid Mech.* 227, 245–272.

Strigle R.F.Jr. 1994. “*Packed Tower Design and Applications*”. Gulf Pub. Co., Houston.

Vafai K. 2000. “*Handbook of Porous Media*”. Marcel Dekker, Inc., New York.

Yamamoto, Y.; M. Potthoff; T. Tanaka; T. Kajishima; and Y. Tsuji. 2001. “Large Eddy Simulation of Turbulent Gas Particle Flow in a Vertical Channel: Effect of Considering Inter-Particle Collision.” *J. Fluid Mech*. 442, 303-334.

Zamankhan, P.; T.Tynjala; W. Jr. Polashenski; P. Zamankhan; and P. Sarkomaa. 1999. “Stress Fluctuations in Continuously Sheared Dense Granular Materials.” *Phys. Rev. E* 60, No. 6, 7149-7156.

THERMALHYDRAULIC MODELING AND ANALYSIS OF CANDU SHUTDOWN COOLING SYSTEM

Ilie Prisecaru and Daniel Dupleac
Power Plant Engineering Faculty
Politehnica University of Bucharest
313 Splaiul Independentei, 060042, sector 6, Bucharest, Romania
E-mail: prisec@cne.pub.ro, danieldu@cne.pub.ro

Niţă Iulian
Center for Engineering and
Technology for Nuclear Objective
Bucharest, Magurele
nitai@router.citon.ro

KEYWORDS
Nuclear engineering, Model development, Model evaluation, Lumped parameter, Continuous simulation

ABSTRACT

The paper presents the mathematical model and thermal-hydraulic analysis of CANDU Shutdown Cooling System (SDCS). The mathematical model for the SDCS equipments was developed. Initially these mathematical models were coded in ACSL simulation language then have been implemented in the Modular Modeling System (MMS) code using the CompGen tools. There are two options for the SDCS operation with different flow path configuration. Essentially, for each option, the thermal regime is time dependent but the flow regime is time independent. Therefore, for the thermal-hydraulic analysis of the system two steps were used. In the first step the flow along the system pipes was determined for each operation stage with the PIPENET code. Subsequently, in the second step, the models developed were used to predict the thermal behavior of reactor core. The results obtained for the cooling of the Primary Heat Transport System (PHTS) from 177°C to 54°C are presented.

INTRODUCTION

After the reactor shutdown, the fuel continues to generate heat by decay of fission products. Normally the initial cooldown of the primary heat transport system (PHTS) is accomplished by using the steam generators. However, at temperatures below 177°C this process becomes ineffective.

The Shutdown Cooling System (SDCS) (CITON, 1990) is provided to remove decay heat from the reactor during shutdown and to cool the PHTS from 177°C to 54°C and hold the system at 54°C for an indefinite period of time. The Candu system consists of a pump and a heat exchanger at each end of the reactor. The design is such that cooldown can be achieved using either the heat transport pumps (flow direction is from inlet headers to the outlet headers via SDC heat exchanger and the SDC pump bypass) or using the shutdown cooling pumps (flow direction is from outlet headers to the inlet headers, see fig. 1) (AECB, 1993). In both cases, the pressure at the inlet headers is sufficient to force water through the core to the opposite outlet headers.

There are two cooldown options for circulating the D2O: by using the Primary Heat Transport Circulating Pumps, or by using the Shutdown Pumps. The initial phase of the two options is similar and involves the use of PHTS pumps to circulate the coolant through the steam generators to lower the PHTS system temperature from 260°C. After the initial cooldown phase, cooldown using PHTS pumps and the SDC heat exchangers can be initiated from a HT system temperature of 177°C (option 1). However, it only reduces the HT system temperature effectively to 121°C with four HT pumps being used or 88°C with two pumps being used. Further cooldown has to be carried out through the use of the SDC pumps and heat exchangers (option 2).

Thermalhydraulic analysis of SDCS must certify that the system is able to remove the decay heat from the fuel following normal shutdown and after certain accident conditions (e.g. feedwater line break). This analysis requires the flow rate and temperature calculation along the systems equipments. However, the physical characteristics of the SDCS show that the momentum and energy balance equations can be decoupled. Consequently, for each system option (different flow path configuration) the flow rate is first calculate. Knowing the flow rate, the temperature distribution can be now evaluated. The paper underlines the SDCS model development and results obtained for the cooling of the Heat Transport System (PHTS) from 177°C are presented.

MODEL DESCRIPTION

Hydraulic model

Different system operation stages have different flow paths. However, the hydraulic resistance of the SDCS loop does not change during system operation for each stage. This fact permits to decouple the momentum balance equation from the energy balance equation. Thus, the flow rates along the SDCS loop can be calculated first for each system operation stage. These calculations are doing with the PIPENET code (Sunrise Systems Limited, 2000). It can performs pipe sizing and pump selection calculations in the steady state, and unsteady flow problems such as water hammer, control systems and hydraulic forces for pipe stress analysis. PIPENET has also a standard module for solving general flow problems with liquids, gases or steam – in pipe and duct networks – cooling water systems, steam distribution systems, HVAC systems.

Figure 1 shows the PIPENET flowchart for option 2 of SDCS configuration, e.g. cooldown has to be carried out through the use of the SDC pumps and heat exchangers.

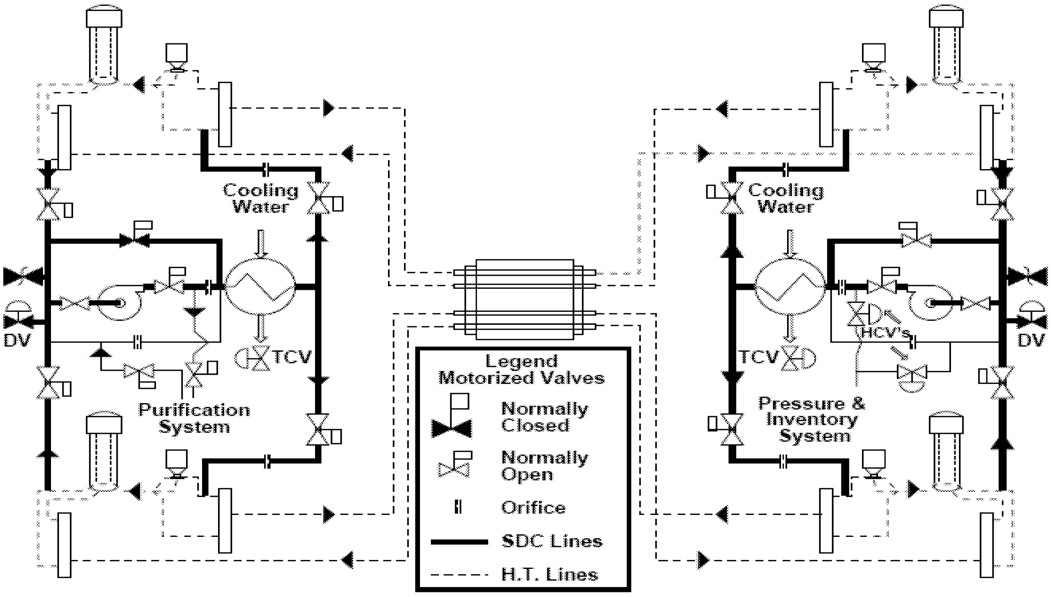


Figure 1: CANDU-6 Shutdown Cooling System

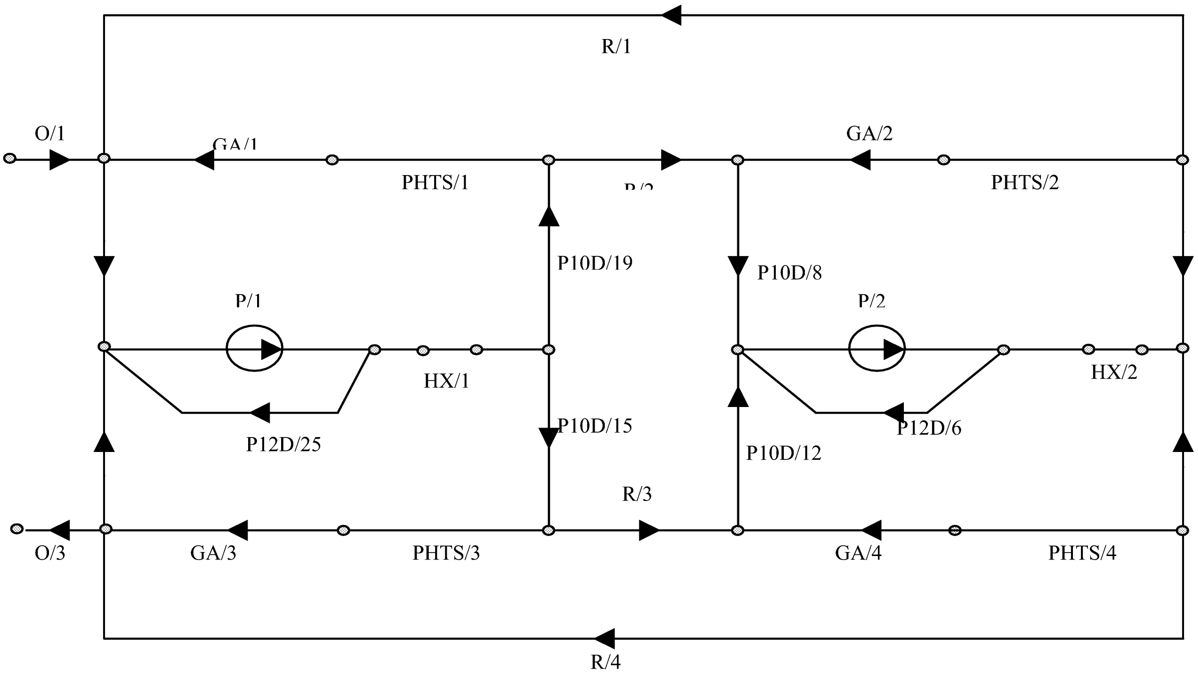


Figure 2: PIPENET Flowchart for Shutdown Cooling System

The flowrate obtained this way is used further in the thermal analysis of the system.

Thermal model

The thermal model consists in modeling the thermal behavior of the following SDCS components: nuclear reactor, steam generators, water-to-water heat exchangers, pumps, and pipes. The analyze scope was to investigate cooling capability of the SDCS in all operating regimes. Consequently all heat sources of the system must be included in the analysis. This includes the reactor decay heat, and stored energy in fuel, pipes and coolant. The heat sinks include the steam generators secondary side (for cooling down from operating condition to 177°C) or the heat exchanger cooling water. The main aspects of the mathematical model for the SDCS equipments are outlined in the following.

Nuclear reactor model

CANDU is a pressurized water reactor developed by AECL in Canada. It uses heavy water in a closed primary circulation loop to transport heat to the steam generators.

Heavy water is heated by the fuel in several small horizontal fuel channels connected by individual feeders to headers.

When the reactor is shut down, modeling of the neutronic behavior is not needed. Thus, the mathematical model of the reactor consists in energy balance equation for the fuel and coolant. The heat sources are represented by the decay heat and the energy stored in the fuel and fuel channels wall. The axial power distribution in fuel channel is supposed to be cosinusoidal as the reactor shutdown did not distort the nominal neutronic flux distribution. For the decay heat an analytical expression given in (Toderas 1993) was used.

The energy balance equation for fuel and coolant are written for a mean fuel channel with the average reactor power and flow rate. Thus, the energy balance equation can be written for the UO$_2$ fuel pellet and fuel clad as:

$$\rho_f c_{pf} \frac{\partial T_f(z,t)}{\partial t} = q'''(z,t) - \frac{2h_{fs}}{r_f}(T_f(z,t) - T_s(z,t)) \tag{1}$$

$$\rho_s \cdot c_{ps} \frac{\partial T_s(z,t)}{\partial t} = \frac{2r_s h_{fs}}{r_f^2 - r_s^2}(T_f(z,t) - T_s(z,t)) - \frac{2hr_s}{r_f^2 - r_s^2}(T_s(z,t) - T_c(z,t)) \tag{2}$$

and for the coolant as:

$$\rho_c c_{pc} \frac{\partial T_c(z,t)}{\partial t} = \frac{hS}{V_c}(T_s(z,t) - T_c(z,t)) - \rho_c \cdot c_{pc} \cdot w_c \frac{\partial T_c(z,t)}{\partial z} + \frac{hS_w}{V_c}(T_w - T_c) \tag{3}$$

However, the fuel clad is thin and has a relatively high thermal conductivity. Thus, a steady-state equation can be used instead of equation (2). Solving for clad temperature from the steady-state form of the eq. (2), and introducing corresponding value in eqs. (1) and (3) we obtain the following set of differential equations:

$$\rho_f c_{pf} \frac{\partial T_f(z,t)}{\partial t} = q'''(z,t) - \frac{2k}{r_f}(T_f(z,t) - T_c(z,t)) \tag{4}$$

$$T_s(z,t) = \frac{T_f + \frac{h}{h_{fs}} T_c}{1 + \frac{h}{h_{fs}}} \tag{5}$$

$$\rho_c c_{pc} \frac{\partial T_c(z,t)}{\partial t} = \frac{kS}{V_c}(T_f(z,t) - T_c(z,t)) - \rho_c \cdot c_{pc} \cdot w_c \frac{\partial T_c(z,t)}{\partial z} + \frac{hS_w}{V_c}(T_w - T_c) \tag{6}$$

where:

$$k = \frac{h_{fs} h}{h_{fs} + h} \tag{7}$$

For the fuel channel wall, the energy balance equation is written as:

$$\rho_w c_{pw} \frac{\partial T_w(z,t)}{\partial t} = -\frac{hS_w}{V_w}(T_w(z,t) - T_c(z,t)) \tag{8}$$

Thus, the thermal model of the nuclear reactor is given by the equations (4), (5), (6) and (8). These equations are solved spatially first by integrating along fuel channel length. As a result a set of ordinary differential equations are obtained. Further, the time integration of these equations provides the temperature behavior of fuel, coolant and fuel channel wall.

Heat exchanger model

The heat exchanger is a tub-shell, water-water type. A lumped parameter model is used for the heat exchanger model (Danila, 1989). The flow is incompressible, thus the low rate is constant along the heat exchanger. The energy stored in the tube wall is neglected. The energy balance equations are written for the heat exchanger outlet temperatures. These are:

$$\frac{dT_t}{dt} = \frac{c_{pt} W_t (T_{ti} - T_{te}) - Q}{\rho_t V_t c_{pt}} \tag{9}$$

$$\frac{dT_{sh}}{dt} = \frac{c_{pt} W_{sh} (T_{shi} - T_{she}) + Q}{\rho_{sh} V_{sh} c_{psh}} \tag{10}$$

where the heat exchanged between fluids is (Frass, 1965):

$$Q = (US)_{HE} \Delta T \tag{11}$$

Only the flow rate is considered to influence the overall heat exchange coefficient.

Pipe model

The main phenomena that must be modeled, from thermal analysis of SDCS point of view, are represented by heat stored in coolant and pipe wall, and the heat loss trough pipe wall. Thus the main pipe model equations are the energy balance equations for the coolant and for the pipe wall. These equations are:

$$\frac{dT_c}{dt} = \frac{W_c (T_{ci} - T_{ce}) c_p - Q}{c_p V \rho_c} \tag{12}$$

$$\frac{dT_w}{dt} = \frac{Q - Q_{loss}}{c_{pw} M_w} \tag{13}$$

The heat exchanged between coolant and pipe wall is calculated similar to the HE model. The heat stored in the pipe wall has a important contribution to the heat sources during the shutdown cooling process.

Steam generator model

The steam generators (SG) are not a SDCS components. However, to simulate the entire cooldown process, starting from the reactor shutdown at nominal condition, a SG model is needed. A detailed SG model was developed for the simulation of the primary or secondary side of the CANDU plant (Prisecaru, 2004). This model is already implemented in the MMS code. Nevertheless, the operational regime of the SG for cooldown process did not require a detailed model. Hence, a simplified model was developed. For regimes involved in cooldown process, the steam generators look like heat source instead of heat sink. A fraction of 55% from coolant flow rate bypasses the reactor core trough SG. As there is not a heat removal on the secondary side of SG, and the pressure of secondary fluid is held constant, the coolant heats up trough SG. Thus, the steam generators became an important heat source after the reactor shut down. The energy balance equations are:

$$\frac{dT_{ce}}{dt} = \frac{W(T_{ci} - T_{ce})}{M_c} - \frac{UA \cdot (\overline{T}_c - T_{sSG})}{M_c c_{pc}} \qquad (14)$$

$$\frac{dT_{sSG}}{dt} = \frac{US \cdot (\overline{T}_c - T_{sSG})}{M_{sSG} c_{psSG}} \qquad (15)$$

In the overall heat transfer coefficient calculation, the tube side heat transfer coefficient is considerate flow dependent whereas the shell side heat transfer coefficient is temperature dependent (Collier, 1994). The energy stored in the SG tube wall was neglected, because this amount is small compared with the energy stored in the secondary side of the SG.

Pump model

From the thermal point of view, the pump looks like a heat source due to frictional forces. Thus, the temperature of the coolant leaving the pump is dependent on pumps power and pump efficiency:

$$T_{ce} = T_{ci} + \frac{P_{pmp}\eta_{pmp}}{W_c c_p} \qquad (16)$$

RESULTS

Initially these mathematical models were coded in ACSL simulation languages (MGA, 1993). The ACSL coded program was used for the validation of the mathematical models. Afterward, the models have been implemented in the Modular Modeling System (MMS) code using the CompGen tools (Framatome Tehnologies, 1998a, 1998b). All the cooldown regimes have been simulated. In the following, the cooldown from the 177°C is presented.

For this simulation the SDCS pumps and heat exchangers models were used. Some of the results are presented in the figures 3 and 4. The coolant temperature at the outlet of heat exchanger are below 85°C from the beginig of simulation even for a flow rate in the heat exchanger shell side of about 40% of nominal value. Hence, the cooldown rate at the outlet of reactor was below prescribed value of 2.8°C/min during all transient. After about 14000 s, temperatures in the SDCS are established to about 35°C at steam generator outlet (reactor inlet) and about 85°C at reactor outlet.

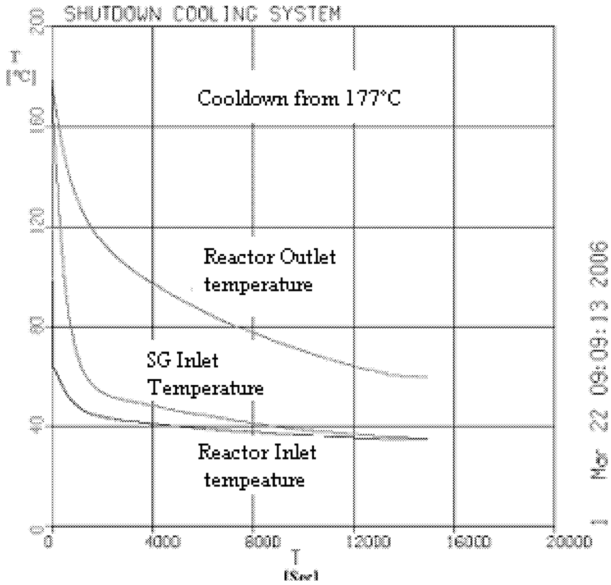


Figure 3: Coolant Temperature at Reactor and SG

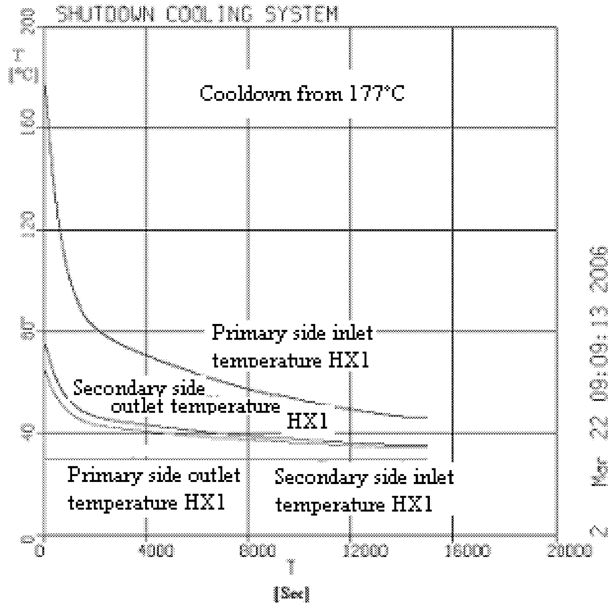


Figure 4: Tube and Shell Side Heat Exchanger Temperature °C

CONCLUSIONS

In this paper mathematical models for the components of the Shutdown Cooling System with application to the Cernavoda N.P.P. are presented. The main aspects of the mathematical model are highlighted. The mathematical models have been coded in ACSL, and after models validation the models are implemented in the MMS code using the CompGen tools. The analyzes were made in porpoise of verifying the capability of Shutdown Cooling System to cool down the Primary Heat Transport System.

Analyze consists in several hydraulic and termohidraulic analyzes. In this paper we presented just a short brief of this regimes. The operating regimes analyzed were the following:

- Hydraulic analyze:
    - initial hydraulic regimes for cool down from 149°C, 177°C, and respectively 260°C to cold state using Primary Heat Transport System (PHTS) pumps;
    - initial hydraulic regimes for cool down from 149°C, 177°C, and respectively 260°C to cold state using Shutdown Cooling System (SDCS) pumps;
    - initial hydraulic regime for cooling down of PHTS drained using SDCS from 77°C to zero power cold state.
- Thermal-hydraulic analyzes:
    - normal operating regimes
        - cooldown regimes using PHTS pumps starting from 149°C, and respectively 177°C;
        - cooldown regimes using SDCS pumps starting from 121°C, 149°C, and respectively 177°C;
        - cooldown regimes using SDCS pumps starting from 77°C, and PHTS drained to zero power cold state.
    - abnormal operating regimes
        - cooldown regimes using PHTS pumps starting from 260°C;
        - cooldown regimes using SDCS pumps starting from 260°C

Analyzing the parameters evolution from the system for all stationary and transient regimes analyzed we observed that all values obtained for thermo hydraulic parameters and time period up to reaching stational regimes of the system are close, in a range of 3%, to technical specifications of design project of the system. Cook down speed was under the value of 2.8°C/min for normal operating regimes. This value was over-passed just in abnormal operating regimes.

REFERENCES

Atomic Energy Control Board (AECB), 1993. *Fundamentals of Power Reactors – Module Two: Nuclear Systems*. Training Manual.

Center for Engineering and Technology for Nuclear Objective (CITON). 1990 *Shutdown Cooling System Technical Description Manual*

Collier, J.G.; and J.R.Thome. 1994. *Convective Boiling and Condensation*, Clarendon Press.

Danila N.; and I. Prisecaru. 1989. "Modeling and simulation of the Heat Exchangers Operation in Dynamic Condition". *Revue Roumaine des scientes techniques*. Serie: Electrotechnique et energetique, nr.3, jullet - spt.

Framatome Tehnologies. 1998a. *MMS Theory Manual Release 5.1.*

Framatome Tehnologies. 1998b. *CompGen Users Manual*.

Frass, A.P.; and M.N. Ozisik. 1965. Heat Exchanger Design, John Wiley & Sons, Inc., New York, London, Sydney.

Mitchell and Gauthier Associates (MGA) Inc. 1993. *Advanced Continuous Simulation Language (ACSL) Reference Manual*. Edition 10.1

Prisecaru, I.; D. Dupleac; and Constantinescu, A. 2004 "CANDU Steam Generators Modeling Using CompGen for MMS Package". *Proceedings of international conference ESMc'2004 (European Simulation and Modelling Conference)*, 25-27 oct, Paris, France

Sunrise Systems Limited. 2000 *PIPENET Users Manual*.

Todreas, N.E.; and M.S. Kazimi 1993, *Nuclear Systems, Thermal hydraulic Fundamentals*,Taylor &Francis.

NOTATION

c_p – heat capacity, J/kg °C
h – heat transfer coefficient, W/m^2 °C
P – power, W
Q – thermal power, W
q''' – power density, W/m^3
r – radius, m
S – surface, m^2
T – temperature, °C
U – overall heat transfer coefficient, W/m^2 °C
V – volume, m^3
W – flow rate, kg/s
w – velocity, m/s
ΔT – mean logaritmic temperature difference, °C
ρ – density, kg/m^3
η - efficiency

Subscript

c – coolant
e – outlet
f – fuel pellet
fs – fuel pellet to clad
i – inlet
pmp – pump
s – fuel clad
sh – shell side
sSG – SG secondary side
t –tube side
w – pipe wall

Solubility of Toxic Compounds from Petroleum Spills into Seawater

M. R. Riazi and Y. M. Al-Roomi
Chemical Engineering Department,
Kuwait University, P. O. Box 5969,
Safat 13060, Kuwait.
E-mail: riazi@kuc01.kuniv.edu.kw

KEYWORDS

Oil spill, seawater, toxic compounds, solubility and dissolution, predictive model.

ABSTRACT

In this paper a mathematical model is presented that can be used to predict the rate and amount of solubility of the most toxic hydrocarbon compounds from a petroleum spill in the seawater. The model can be applied to crude oil as well as its products floating on seawater. The information required are oil specifications, initial spill area and volume, air and water temperature and the wind speed.

INTRODUCTION

Due to increase in the rate of oil transportation through sea and off shore oil production activities there has been an increase in crude oil (or its products) oil spills occurring on seawater surface (ASCE Task Force 1996). Presence of oil spills floated on seawater surface causes significant environmental, ecological and economical damages to the surrounding areas due to contamination of water with hydrocarbons and pollutant materials (Kuiper and Van den Brink 1987).

Oil spills may go through various dynamic processes both at the seawater surface and in water (Green and Trett 1989).

On the water surface the processes that may occur are:
Evaporation due to sun radiation
Oxidation due to energy from sun and oxygen from air
Spreading due to interfacial tension with water
Emulsification due immiscibility with water

Inside water below seawater surface the following processes may occur:
Dispersion
Biodegradation
Dissolution
Sedimentation to the bottom of sea ground due
to the presence of compounds heavier than water.

A comprehensive model that considers all these dynamic processes is not reported in the literature but there are many models that consider some major processes (Spaulding 1988, Stiver et al. 1989, Villoria, et al. 1991, Riazi and Edalat 1996, Riazi and Alenzi 1999). These researchers have reviewed and suggested different models for the fate of oil spills.

The rate at which a hydrocarbon dissolves in water is generally lower than the rate of evaporation under the same conditions (Riazi and Edalat 1996, Riazi and Alenzi 1999). It is widely considered that, after volatility, the most significant property of oil components, from the point of view of their behavior in aquatic environments, is their solubility in water (Wheeler 1987). It has been shown that the rate of oil dissolution in water is small in comparison with rate of oil evaporation and usually the amount of oil dissolved is less than 1% of original mass of the spill (Riazi and Edalat 1996). But the dissolved concentrations of hydrocarbons in water concern from a toxicological viewpoint and it is important to know the exact amount of oil dissolved in water as a result of an oil spill. Aromatic hydrocarbons especially mono-aromatics such as benzenes are the most toxic compounds and their amount in water determines the degree of toxicity in water. The physical process of dissolution is well understood, but the description in the case of oil spills is complicated, due to the complex oil composition with hundreds of components and the necessity of describing the dissolution of a single component with component-specific parameters. The component-specific description may be necessary because toxicity is component-specific as well. The most soluble oil components are usually the most toxic. Even low concentrations of these toxic compounds could lead to serious effects on biological systems.

In this paper the model previously developed (Riazi and Alenzi 1999) is modified along with characterization methods available for crude oil and petroleum fractions (Riazi 2005) to predict the rate at which toxic compounds mainly aromatics and mono-aromatics are dissolved in water. The model may be used in simulators that are used to simulate environmental damages due to oil spills. Such information may also be used in better selection of a method for cleaning operations after occurrence of an oil spill.

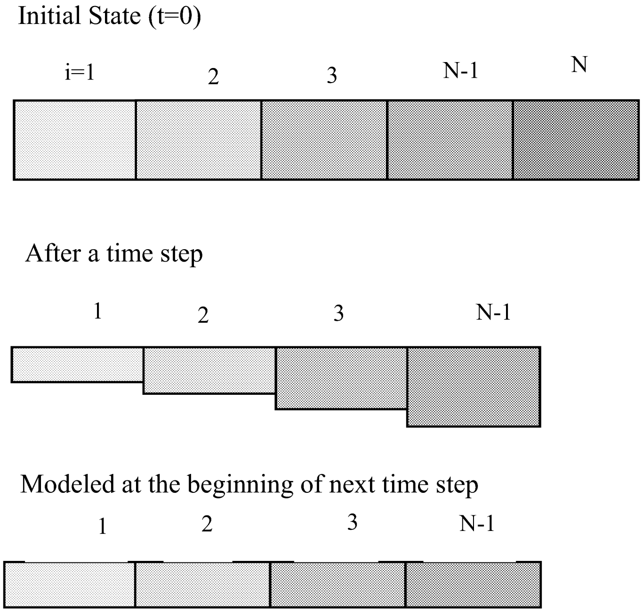


Figure 1. Modeling Scheme for a Crude Oil Spi

MODEL DESCRIPTION

Summary of model description is shown in Figure 1 and the calculation procedure is outlined in Figure 2. In the model it is assumed that the whole surface area of the spill is divided into N areas each corresponding to a pseudo-component. Thickness of this oil segment varies with time. Following the analytical relations previously developed (Riazi and Al-Enezi 1999), and using a semi-analytical approach one can calculate the amount of each component dissolved in water versus time. Each pseudo-component is divided into three sub-components from paraffins, naphthenes and aromatics. In addition the aromatic portion is divided into two parts of monocyclic compounds such as benzenes and toluenes and polycyclic aromatics. Benzene compounds and in general mono-aromatics are the most toxic compounds in a petroleum mixture.

The rate of evaporation of component “i” is proportional to its vapor pressure (P_i^{vap}) through a proportionality constant shown by K_i^{vap} and similarly the rate of dissolution of component “i” is proportional to its solubility in water (C_{si}) through another proportionality constant shown by K_i^{dis}. Volume fraction of component “i” vaporized (F_{Vi}^{vap}) or dissolved in water (F_{Vi}^{dis}) and the area A_i are calculated through the following relations for each time step of Δt.

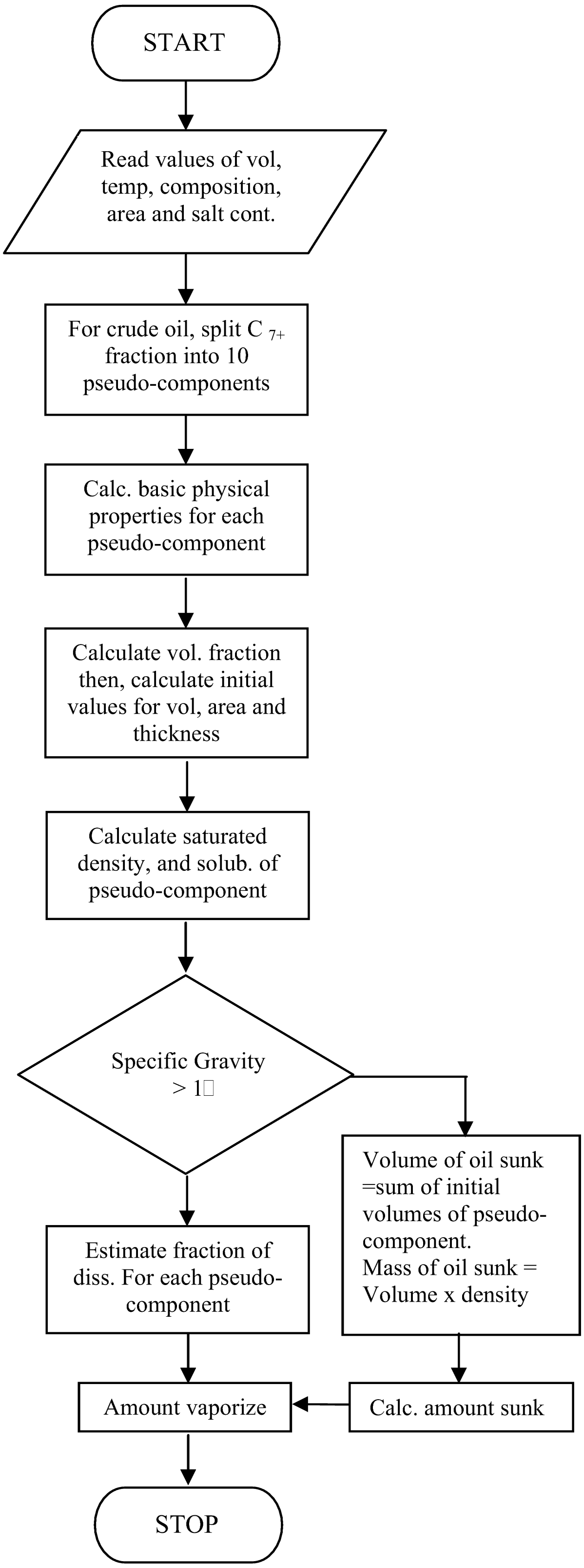

START
Read values of vol, temp, composition, area and salt cont.
For crude oil, split C 7+ fraction into 10 pseudo-components
Calc. basic physical properties for each pseudo-component
Calculate vol. fraction then, calculate initial values for vol, area and thickness
Calculate saturated density, and solub. of pseudo-component
Specific Gravity > 1
Volume of oil sunk =sum of initial volumes of pseudo-component. Mass of oil sunk = Volume x density
Estimate fraction of diss. For each pseudo-component
Amount vaporize
Calc. amount sunk
STOP

Figure 2. Summary of calculations scheme

$$F_{Vi}^{vap} = 1 - \exp(-Q_i^{vap}\Delta t) \quad (1)$$

$$F_{Vi}^{dis} = 1 - \exp(-Q_i^{dis}\Delta t) \quad (2)$$

$$A_i = A_{io}\exp(-Q_i^{vap}\Delta t) \quad (3)$$

where parameters Q_i^{vap} and Q_i^{dis} are defined as follows

$$Q_i^{vap} = \frac{K_i^{vap} Z_{liq,i}^{sat}}{y_i} \quad (4)$$

$$Q_i^{dis} = \frac{K_i^{dis} C_{si}}{y_i \rho_{mi}} \quad (5)$$

$$Z_{liq,i}^{sat} = \frac{P_i^{sat} M_i}{\rho_{liq,i}^{sat} RT} \quad (6)$$

where ρ_{mi} is liquid molar density while $\rho_{liq,i}^{sat}$ is the absolute saturated liquid density of component "i" at temperature T. C_{si} is the molar solubility of component "i" in water at water temperature. The following relations (Riazi and AlEnzi 1999) were developed for estimation of K_i^{vap} and K_i^{dis} based on laboratory data for several narrow-cut petroleum fractions obtained from Kuwait National Petroleum Company (KNPC).

$$K_i^{vap} = 1.5 \times 10^{-5} U^{0.8} (T / M_i)^2 \quad (7)$$

$$K_i^{dis} = \frac{4.18 \times 10^{-9} T^{0.67}}{V_{Ai}^{0.4} A_i^{0.1}} \quad (8)$$

in which K_i^{vap} and K_i^{dis} are in m/s, T is in K, V_{Ai} is the molar volume of component "i" at its normal boiling point in m^3/gmol. A_i is the surface area of component "i" in m^2. M_i is the molecular weight of "i" and U is the wind speed in m/s.

Solubility of component "i" in water (C_{si}) is calculated from the following relation in terms of molecular weight, temperature and salt concentration in water:

$$\begin{aligned} C_{si} = \exp[(4.6 - 0.0036M_i) + \\ (0.1 - 0.0018M_i)S_w - 4250/T] \end{aligned} \quad (9)$$

in which C_{si} is in mol/L, T is the absolute temperature of water in K and S_w is the salt concentration in water in weight%.

Once volume fraction of each component vaporized or dissolved is calculated through eqs. 6 and 7, mass of each component vaporized (m_i^{vap}) or dissolved (m_i^{dis}) may be calculated through the following relations:

$$m_i^{vap} = \rho_{liq,i}^{sat} F_{Vi}^{vap} V_{oi} \quad (10)$$

$$m_i^{dis} = \rho_{liq,i}^{sat} F_{Vi}^{dis} V_{oi} \quad (11)$$

Total mass of spill disappeared after time t is then calculated through sum of mass disappeared by evaporation, dissolution and sedimentation:

$$m_i = m_i^{vap} + m_i^{dis} + m_i^{sed} \quad (12)$$

Mass of oil remaining on the seawater surface after time t is calculated from:

$$m = \sum_{i=1}^{N} (m_{oi} - m_i) \quad (13)$$

where m_{oi} is the initial mass of component i.

To find the amount of toxic compounds we use the following equations to break the oil mixture into mono-aromatics and polyaromatic compounds as discussed through the methods given in the ASTM manual 50 (Riazi 2005).

$$x_{MA} = -62.8245 + 59.90816R_i - 0.0248335m \quad (14)$$

$$x_{PA} = 11.88175 - 11.2213R_i + 0.023745m \quad (15)$$

$$x_A = x_{MA} + x_{PA} \quad (16)$$

In the above relations X_{MA}, X_{PA}, and X_A are the weight fractions of monoaromatics (MA), polyaromatics and aromatics (A), respectively. Parameters R_i and m are defined as follows:

$$R_i = n_{20} - d_{20} / 2 \quad (17)$$

$$m = M(n_{20} - 1.475) \quad (18)$$

where n20 and d20 are refractive index and density of oil components at 20 °C. These parameters can be accurately estimated from oil characterization data as shown in the ASTM Manual 50 (Riazi, 2005). For crude oil samples and wide heavy oil products such as fule oils and residues the following distribution model can be used to split the mixtures into a number of pseudocomponents with known characterization data (Riazi, 1997, 2005).

$$F(P^*) = \frac{B^2}{A} P^{*^{B-1}} \exp(-\frac{B}{A} P^{*^B}) \quad (19)$$

$$P^* = \frac{P - P_o}{P_o} \quad (20)$$

where F is the probability function for a property P such as molecular weight and x is the cumulative mole fraction. A, B and P_o are constants for each property and each oil sample.

Model predictions for a Kuwait crude oil with specific gravity of 0.891 and molecular weight of 267 are shown in Figure 3 at temperature of 40 C and Figure 4 at temperature of 20 C. The initial mass of oil spill was 450 g and the maximum aromatics that can be dissolved is 0.4 g while the amount of mono-aromatics would be about 0.2 g after a week. The model is useful to determine and monitor the concentration of toxic compounds dissolved in seawater as a results of oil spill occurrence. The model is convenient to apply in simulators developed to evaluate environmental damages from oil spills into aquatic environments.

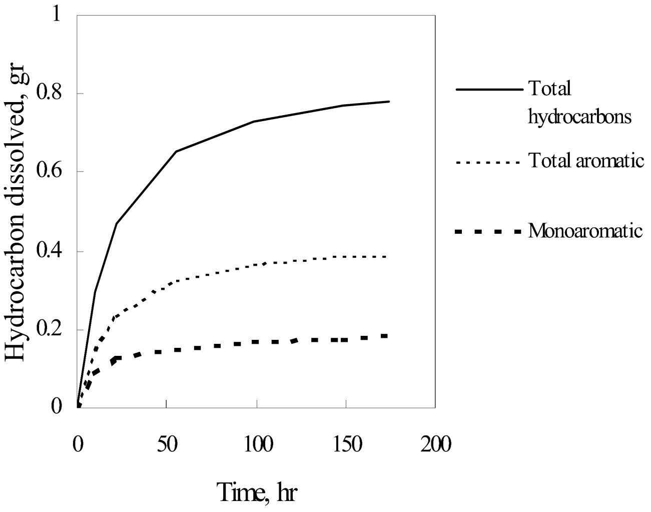


Figure 3. Model prediction for the rate of dissolution of a crude oil spill at 40 °C

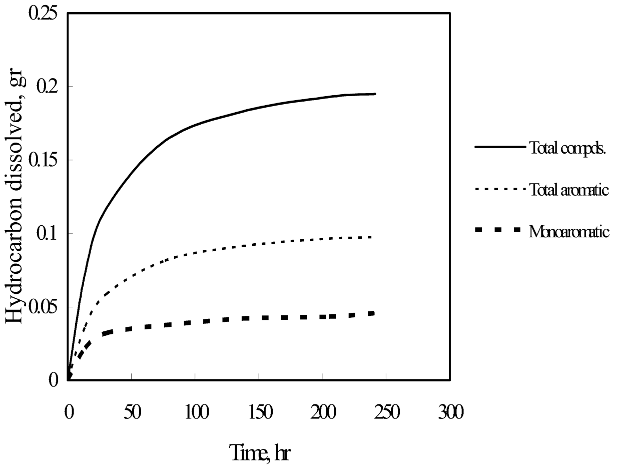


Figure 4. Model prediction for the rate of dissolution of a crude oil spill at 20 °C

ACKNOWLEDGEMENT

The funding for this project was provided by Kuwait University Project No EC05/05 and is appreciated.

NOMENCLATURE

A = Oil spill surface area
A_o = Initial oil spill surface area
C_s =Oil solubility in water (gmol/L)
d = Liquid density at 20 C and 1 atm, g/cm^3.
F_V = Volume fraction of oil disappeared, dimensionless
K^{dis} = Mass transfer coefficient for the rate of dissolution, m/s
K^{vap} = Mass transfer coefficient for the rate of evaporation, m/s
M = Molecular weight, g/gmol
m = Parameter defined by eq. 18.
n = Sodium D-line refractive index at 20 C and 1 atm.
P^{sat} = Saturation pressure of oil component at temperature T
Q^{dis} = Parameter defined in eq. 5, s^{-1} or day^{-1}
Q^{vap} = Parameter defined in eq. 4, s^{-1} or day^{-1}
R = Gas constant
R_i = Refractivity intercept defined by eq. 17
S_w= Salt concentration in seawater, weight %
T = Temperature, K
t = Time, s
V = Oil spill volume, m^3
V_A = Oil molar volume at normal boiling point, cm^3/gmol
V_o = Initial oil spill volume, m^3
U = Wind speed, m/s
X_A = Weight fraction of aromatics in a hydrocarbon group
X_{MA} = Weight fraction of mono-aromatics in a hydrocarbon group
X_{PA} = Weight fraction of poly-aromatics in a hydrocarbon group
y = Oil spill thickness
Z_{liq}^{vap} = Saturated liquid compressibility factor, dimensionless, defined by eq. 6

Greek letters:
ρ_{mi} = Oil component liquid molar density, gmol/m^3
ρ_{liq}^{sat} = Saturated liquid density
Δ t = Time, day, hour or sec.

Subscripts:
i = Pseudocomponent “i”
o = Initial values (values at t = 0)
20 = value of a property at 20 C

REFERENCES

ASCE Task Force. 1996. “State-of-the-Art Review of Modeling Transport and Fate of Oil Spills”, *J. Hydraulic Engineering*, (Nov.), 594-609.

Green, J. and M.W. Trett. 1989. *"The Fate and Effects of Oil in Freshwater"*, Elsevier, London.

Kuiper, J. and W.J. Van den Brink. 1987. *"Fate and Effects of Oil in Marine Ecosystems"*, Martinus Nijhoff Publishers, Boston.

Riazi, M. R., 1997. "A Continuous Distibution Model for C_{7+} Fraction Characterization", *Ind. Eng. Chem. Research,* 37, 4299-4307.

Riazi, M.R. 2005. "Characterization and Properties of Petroleum Fractions", ASTM Manual 50, *ASTM International*, Conshohocken, PA ., website: http://www.astm.org/mnl50.htm

Riazi, M.R. and G. Alenzi, G. 1999. "A Mathematical Model for the Rate of Oil Spill Disappearance from Seawater for Kuwaiti Crude and Its Products", *Chem. Eng. Journal*, 73, 161-172.

Riazi, M.R. and M. Edalat. 1996. "Prediction of the Rate of Oil Removal from Seawater by Evaporation and Dissolution," *J. Pet. Sci. and Eng.*, 16, 291-300.

Spaulding, M.L. 1988. "A State-of-Art Review of Oil Trajectory and Fate Modeling", *Oil and Chemical Pollution*, Vol. 4, 39-55.

Stiver, W., W. Y. Shiu and D. MacKay. 1989 "Evaporation times and rates of specific hydrocarbons in oil spills", *Environ. Sci. Technol.*, Vol. 23, No. 1, 101-105.

Villoria, C.M., A. E. Anselmi, S. A. Intevep and F. R. Garcia, F.R. 1991. "An Oil Spill Fate Model", *SPE23371*, 445-454.

Wheeler, R.B. 1987. *"The Fate of Petroleum in the Marine Environment"*, Production Research Company Special Report, Exxon, New Jersey .

BIOGRAPHY

M. R. Riazi is currently a professor of chemical engineering at Kuwait University. Previously he was an assistant professor of chemical engineering at Pennsylvania State University where he also received his M.Sc. and Ph.D. degrees. He was also visiting professors at the following universities: Trondheim (Norway), Illinois (Chicago, U.S.), Texas (Austin, U.S.), Wright State (Dayton, U.S.) and McGill (Montreal, Canada). He is an associate editor of Journal of Petroleum Science and Engineering (Elsevier), regional editor of World Review for Science, Technology and Sustainable Development (Inderscience, U.K.) and editorial board member of Journal of ASTM International (ASTM, U.S.). He is the author and coauthor of about 70 refereed papers in international journals, 30 conference papers, 5 book chapters and sole author of two books. He was awarded a diploma of honor by the American Petroleum Association. His recent book on petroleum properties: (http://www.astm.org/mnl50.htm) is in worldwide use by the petroleum industry. His email address is: riazi@kuc01.kuniv.edu.kw

Yousef Al-Roomi is currently an associate professor of chemical engineering at Kuwait University. He was previously chairman of chemical engineering department where he has been involved in research and teaching in the area of molecular simulation and processing. He received his B.Sc. in applied chemistry from University of Missouri (Columbia, U.S.), M.Sc. in chemical engineering from University of Columbia (New York, NY) and Ph.D. in Chemical Engineering from Cornell University (Ithaca, New York). He has widely published in the areas of polymer, water, petroleum and chemical technology in addition to consultation for the industry. His email is: yalroomi@kuc01.kuniv.edu.kw

AI-BASED SIMULATION METHODOLOGY

GAME ANALYSIS BY MEANS OF SIMULATION

Roland Angerer
3united
Jakob-Haringer-Straße 5A
5020 Salzburg, Austria
e-mail: r.angerer@3united.com

Helge Hagenauer
FB Computerwissenschaften
Universität Salzburg
Jakob-Haringer-Straße 2
5020 Salzburg, Austria
e-mail: hagenau@cosy.sbg.ac.at

ABSTRACT

The development and research of traditional games is still based on either observations obtained by playing a game or, a time-consuming mathematical analysis. This paper proposes the use of computer simulation to aid in game development and research. Using a simulation framework especially developed for this purpose, players with certain strategies are simulated to play a given game. Statistics are obtained that lead to valuable insights about the properties of a studied game and might help a game designer to further improve his prototype.

INTRODUCTION

This paper will focus on traditional games (e.g. board games, card games, . . .), defined by rules and the needed playing-material (e.g. game pieces, cards, board, . . .). These games are as old as mankind itself with some of the oldest games like the "Royal Game of Ur" or "Senet" dating back to 3000BC.
The development of games is traditionally based on prototyping: ruleset and playing-material are evaluated and then modified to resolve the encountered problems. A given prototype is evaluated by testers, which normally play the game until they find problems or decide that the game is ready to be published. Therefore this traditional method of game analysis greatly depends on the situations covered during the tests and the analytical skills of the involved players. As this method clearly has its drawbacks (which lead to many flaws in published games) we shall examine more advanced methods for game analysis:

Game Theory was introduced by John von Neumann and Oskar Morgenstern in 1944 to study strategic situations, where players choose different actions in an attempt to maximize their returns. The methods introduced with game theory are mostly used in military and economical applications, where the given maximization problems are reformulated into games.

Game Design Patterns were introduced by the "Game Design Patterns" project, which tried to find certain design patterns used in many modern games. These patterns may be used to develop new games or draw conclusions about existing games by finding the used patterns.

Unfortunately both approaches are not very suitable for evaluation of a prototype. Game theory tries to answer the question of winning percentage and provides methods to find optimal strategies. Design Patterns on the other hand are too general to find flaws in concrete games.
The goal of this paper is to introduce simulation as a new method for game analysis suitable to aid in the development process and game research in general. We will discuss the great potential of simulation to analyze games as well as the boundaries to its application.

The paper is organized as follows: the next chapter describes the basic principles of simulating traditional games and gives an overview of the gsimj framework followed by a short example sketching some aspects for using it. Then some results on simulated games are presented and discussed. The conclusions sum up the paper.

GAME SIMULATION

Simulation can be classified as *deterministic* or *stochastic*. Traditionally, games include random events like rolling dices or drawing cards from a shuffled stack and have to be classified as stochastic. Even with games like chess which do not contain random elements themselves you have to use stochastic elements in the strategies to produce different results at each simulation run.
Another useful distinction is between *continuous* and *discrete* simulation. As state changes in a game normally occur at discrete positions in time, a continuous simulation of time is not necessary. While more traditional simulations use a natural time-based scheduling unit like seconds or minutes, games are best described and simulated by using a players move for scheduling.
As we have seen most games can be described by a discrete event simulation. There are two broadly used implementation techniques for this kind of simulation: *event scheduling* and *process interaction.* A quite nat-

ural way of modeling games would be to define the following entities as processes:

Game is holding the state information for the game. It activates players when they should make a move or have to act according to the rules of the game. Access to the gaming material respecting the given rules is also provided by this entity. The game could also be used to model a controlled interaction between players.

Player represents a player and his state information such as playing material currently under his control. A player provides all the actions the game may want him to perform and therefore defines the abilities and responsibilities of a player.

It is convenient to define another entity: **Strategy**. The strategy is used whenever the player has to make a decision and therefore defines the way the player is trying to win, but not the way the game is played. The game-play is defined by the interfaces of the processes for game and player and the interactions defined by these processes. Typically a game will activate each player joined in the game to make his move until a condition ending the game is met. With most games this condition is a player winning the game according to its rules.

gsimj – A Framework for Game Simulation

While most existing simulation frameworks concentrate on the simulation time and provide a wealth of tools and statistics for time-related issues game simulation does not focus on time-related questions. Furthermore most simulation frameworks do not provide statistical elements that are suitable for analysis of a large number of simulation runs. With game simulation meaningful statistics can only be retrieved by playing and therefore simulating the game for a reasonable large amount of plays.
So we decided to develop our own simulation framework: gsimj – **g**ame **sim**ulation in **J**ava. Its main purpose is to speed up the development of game simulations by providing the main components for games and their basic interaction. The framework is structured into the following packages:

- gsimj – Main package, containing the components generally used for game simulation: a game, its players and their strategies.

- gsimj.gui – Any components belonging to the graphical user interface provided by gsimj are part of this package.

- gsimj.stats – This package represents the components used to gather and display statistical information.

- gsimj.util – Components which might be useful for some game simulations are collected in the gsimj.util package.

IMPLEMENTATION OF A GAME

Conceptually the implementation of a game model is done by subclassing the three main components in gsimj: Game, Player and Strategy.
We want to demonstrate the usage of gsimj by the example of the well known game Tic Tac Toe. A Game is initialized with the number of turns after which the game should stop. Tic Tac Toe has a natural maximum of nine moves, but the first player is able to win the game after just five moves. Therefore we need to override the Game.**isFinished** method to verify if the game is already won by checking all possible win situations:

```
public boolean isFinished() {
   for (each win situation) {
      if (currentState = WinSituation) {
         return true;
      }
   }
   // check maximum moves
   super.isFinished();
}
```

After the game is started it will call each Players **move** method until a player has won or the maximum number of moves are reached. Therefore we have to implement the abstract **move** method so the players actually choose a spot on the board. To separate the mechanism from the policy we will let a Strategy component decide about the spot. This seems like a lot of overhead for Tic Tac Toe, but makes great sense for more complex games.

```
public boolean move() {
   Game game = this.getGame();
   Strategy strategy = this.getStrategy();

   // get pick from strategy
   int pick = strategy.getPick(game.board);
   // submit decision to the game
   game.pick(pick, this);
}
```

We submitted the current state information to the players strategy as a parameter. This concept is practical for smaller games while with more complex gaming material the strategy will access the game directly.
Finally we have to define a starter application, that will bring up the simulation GUI and let us choose the number of players, their strategies and other additional parameters:

```
SimulationUI ui = new SimulationUI(
   TicTacToeGame.class,
   TicTacToePlayer.class
);
ui.addStrategy(RandomStrategy.class);
ui.addStrategy(AdvancedStrategy.class);
ui.setPlayerCount(2);
// setExecutionCount(min, default, max)
```

```
ui.setExecutionCount(1, 100, 1000);
ui.show();
```

SIMULATED GAMES

In order to prove the use of simulation for game analysis it is necessary to simulate different games and verify the obtained results. Therefore we decided to implement three games with different complexity with gsimj: Tic Tac Toe, Cluedo and Settlers of Catan. As expected the more complex a game becomes, the more effort is needed to model it. Especially the gaming material and strategies for Settlers of Catan—the most complex game—were very time-consuming.
However the performance of the simulation runs was acceptable, although the simulation framework and its components are not yet optimized for speed. For example a simulation of 1,000 executions of Settlers of Catan with four players does only take from one to five minutes on average hardware[1].
In the following we explain game specific design aspects and show some interesting simulation results for Tic Tac Toe and Settlers of Catan.

Tic Tac Toe

Maybe the most interesting part of Tic Tac Toe is the strategy that is applied to win the game. We created the following strategies to choose from:

random features a first-time player. When playing for the first time many players choose their moves randomly. Unlike many normal players this strategy does not check if it could win with its next move.

advanced represents a player with some experience. The player has the following objectives, ordered by their priorities:

1. connect three in a row
2. keep the opponent from connecting three in a row
3. connect two in a row

When the player tries to connect two he chooses the position with the most winning situations. Although this strategy is quite sophisticated it does not guarantee a draw. The advanced player is not able to defend against two possible completions in two rows.

statistical is an attempt to make the best move by analyzing all possible endings for each move. Every choice at a given move is rated by the number of wins compared to the total number of possible endings. Although this approach favors choices with a high statistical possibility to win it is far from a perfect strategy.

expert represents an expert player who will not lose the game. This strategy defines the best way the game can be played using a minimax algorithm introduced by John von Neumann.

	random	advanced	statistical	expert
random	58.5/28.6	1.7/91.2	17.3/82.7	0.0/76.5
advanced	99.0/ 0.0	9.9/ 0.0	100.0/ 0.0	0.0/ 0.0
statistical	98.8/ 0.0	0.0/ 0.0	100.0/ 0.0	0.0/ 0.0
expert	96.4/ 0.0	47.4/ 0.0	100.0/ 0.0	0.0/ 0.0

Table 1: Tic Tac Toe strategy-evaluation

Table 1 shows an evaluation of all strategies against each other. The rows represent the first players strategy while the columns display the second players strategy. All cells show the first and second players win percentage.
As you can see, the statistical strategy is not competitive at all. While this might seem odd at first, it becomes quite obvious: even if there are a lot of good endings for a choice, one bad ending is enough to make the player lose.
Another contra-intuitive observation is, that the expert strategy does not beat every other strategy in respect to the win percentage. For example the advanced and statistical strategies outperform the expert against a random player. Reason is the experts implementation: The used minimax algorithm tries to minimize the opponents chances, considering him to make the best possible move to maximize his own chances.

Settlers of Catan

Settlers of Catan is a very successful German board game. Although the basic rules can be summarized in one page the flexible board layout, trading and building mechanisms lead to a complex game. Many additional extensions and a wealth of scenarios introduce further elements to the game-play.
One of the most demanding tasks was to implement the data-structure for representing the hexagonal board layout for Settlers of Catan. To make matters even more interesting the data structure has to be able to handle not only the hexagonal pieces, that make up the board, but also the nodes and edges between those pieces.
To be able to address the hexagonal fields in a reasonable way we transformed the hexagonal layout into a two-dimensional array by shifting every second line for a half hexagonal distance (see also figure 1). Calculating the coordinates of the hexagons surrounding the current one needs some attention as well: While this is simple for the left and right neighbours, when going up or down one row the according columns depend on whether the current row has an even or odd number and the direction

[1] The reference system was a 1.8 GHz pentium with 1 GB memory

(e. g. above left or above right). Figure 2 illustrates the calculation for the upper left and right hexagons.

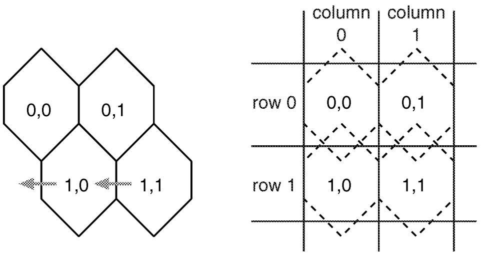


Figure 1: Layout transformation

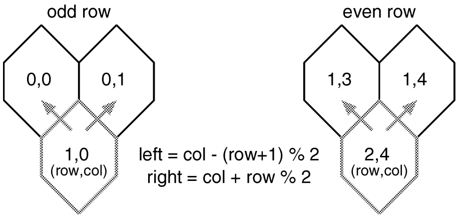


Figure 2: Hex coordinate computation

We decided to address the nodes (cities) and edges (roads) relative to their adjoining hexagons and therefore connected the resulting graph once at simulation startup.

Unlike Tic Tac Toe and Cluedo there is no such thing as a "perfect strategy" for Settlers of Catan. Although the options provided by the game-play are much too complex to be able to create the best strategy, we are able to identify three basic decisions the strategy has to cope with:

- Where to build?
- What to build/buy?
- When and what to trade?

The used strategy is heavily relying on ratings to decide where it will build. In a quite natural approach the rating for a spot is derived from the number of possible dice-combinations for its surrounding fields. Generally the strategy tries to win by building settlements and upgrading them to cities. It chooses the appropriate action based on the resources the player posseses. Finally the trading is based on the decision for "what to build?". The resources of the player are separted into needed and tradeable resources and the strategy requests bids from every player, choosing the one with the fewest requested resources.

Table 2 shows the players *relative winning rates*[2] and the average amount of moves to finish the game after 1,000 simulation runs of Settlers of Catan. Four different board layouts were used: `default.board` represents the board suggested by the games manual, `random.board` chooses the positions of the hexagons at random, `unfair.board` tries to favour the first player over the others and `islands.board` where Catan consists of three islands.

	Player 1	Player 2	Player 3	Player 4	avg. moves
default.board					
3 players	1.0	0.98	1.01	–	58.64
4 players	1.0	0.97	0.99	1.03	57.12
random.board					
3 players	1.01	1.04	0.95	–	74.49
4 players	1.06	0.92	1.02	1.0	74.9
unfair.board					
3 players	1.0	1.01	1.0	–	56.73
4 players	1.01	1.02	0.98	0.99	64.5
islands.board					
3 players	0.96	1.03	1.02	–	71.64
4 players	1.04	1.01	0.98	0.96	87.96

Table 2: Settlers of Catan board comparison

These results allow us to draw two conclusions: First the game is extremely well balanced in respect to different board layouts, because the winning rates of all players are very near to their expected winning rates. Secondly we can see, that the position of the player (first or forth) does not impact his chances to win the game.

CONCLUSION

We believe that the non-trivial results obtained for the simulated example games show the relevance and benefits of the proposed approach to game development as well as research of games and their strategies. Simulation can be used where the more traditional methods of game analysis have reached their limitations due to the complexity of modern games.

Although the benefit of simulation for game analysis is obvious, one factor needs serious consideration: The time and effort needed to implement a given game. A not so complex game can be implemented with very little effort and almost instantly presents interesting results. However, a complex game like Settlers of Catan introduces a lot of challenges. Primarily, these are the representation of the playing material and game mechanisms (like trade) as well as the implementation of a realistic strategy, that is able to exploit any shortcomings in the rule-set and therefore make them visible in the gathered statistics.

In contrast to the games implementation the extraction of statistics is rather easy. Gathering additional statistics is done by adding the corresponding statistical

[2] actual winning rate divided by the expected winning rate (with equal distribution)

component to the game and calling its **update** method at a suitable location. The framework takes care of the computation and display of the gathered information.

Outlook

Based on the already mentioned Game Design Patterns reusable objects as well as mechanisms should be implemented in the gsimj framework. Thereby the development of a game simulation would be greatly simplified to identifying the used Game Design Patterns and configuring the corresponding OO-objects.
Another very interesting task would be to actually accompany the development of a game by simulation rather than analyzing an already published game. How much would the development process be able to benefit from the described approach to game analysis? How much implementation effort for the simulation would be needed in comparison to the total game-development?

REFERENCES

Game Design Patterns Project, 2005. *Game Design Patterns.* `http://www.gamedesignpatterns.org/`, accessed on 2005–09–12.

Page B., 1991. *Diskrete Simulation — Eine Einführung mit Modula-2.* Springer-Lehrbuch.

Ross D., 2005. *Game Theory.* In E.N. Zalta (Ed.), *The Stanford Encyclopedia of Philosophy.* `http://plato.stanford.edu/archives/win2005/entries/game-theory/`, accessed on 2006–02–11.

Tzeng C.H., 1988. *A Theory of Heuristic Information in Game-Tree Search.* Springer-Verlag.

von Neumann J. and Morgenstern O., 1944. *The Theory of Games and Economic Behavior.* Princeton University Press.

Vorobjoff N.N., 1967. *Grundlagen der Spieltheorie und ihre praktische Bedeutung.* physica, 2nd ed.

Weintraub E.R. (Ed.), 1992. *Toward a History of Game Theory.* Duke University Press.

SIMULATION-BASED OPTIMISATION USING GLOBAL SEARCH AND NEURAL NETWORK METAMODELS

Anna Persson
Henrik Grimm
Amos Ng
Centre for Intelligent Automation
University of Skövde
Box 408, Skövde
Sweden
E-mail: {anna.persson, henrik.grimm, amos.ng}@his.se

KEYWORDS
Optimisation, Simulation, Global Search, Metamodel,
Neural Network

ABSTRACT

This paper presents a new population-based metaheuristic algorithm for simulation-based optimisation. The proposed algorithm uses metamodels for efficiency enhancement. Similar to other population-based metaheuristics, such as Genetic Algorithms (GA), it generates and maintains a population of solutions and progresses incrementally generation by generation using genetic operations. The difference is that a trained metamodel is used to discard inferior candidate solutions while keeping the most promising ones. This could significantly enhance the efficiency of the optimisation process by avoiding time-consuming simulation runs for the candidate solutions that lack potential. During the optimisation, the accuracy of the metamodel is constantly improved as on-line training is applied after each generation of solutions have been simulated. The proposed algorithm is implemented on a benchmark optimisation problem and initial results show that the algorithm is able to effectively enhance the performance of the simulation-based optimisation process in comparison with a standard GA-based approach.

1 INTRODUCTION

One of the most important and challenging subjects in the simulation field today is simulation-based optimisation (Buchholz 2005). It has shown to be a powerful technique for systems improvement and has been successfully applied to address a wide range of real-world industrial optimisation problems (April et al. 2004). The general problem in simulation-based optimisation (SO) is to find a setting of decision variables that maximize or minimize a given objective function, assuming that it cannot be computed analytically but must be estimated through simulation.

Population-based optimisation methods such as Genetic Algorithms and Evolutionary Strategies are powerful search algorithms commonly found in SO. These algorithms are increasingly being used to solve a wide range of different optimisation problems, especially when there are a large number of parameters with complex dependencies. However, the main weakness of using population-based optimisation methods in SO is that they require a large number of fitness evaluations. Typically, a population-based optimisation strategy requires thousands of simulation evaluations and it is not uncommon for simulation models to run for hours. For practical applications of SO it is of critical importance that the optimisation process is constrained within reasonable time limits and the efficiency of the optimisation process is crucial.

One potential way to enhance the efficiency of SO and reduce the number of time-consuming simulation runs is to employ computationally cheap metamodels (Alam et al. 2004). Metamodels, also known as surrogate or approximate models, are essentially a “model of the model” which may be used to approximate a simulation model. By adopting metamodels, the computational burden of the optimisation process can be greatly reduced since the computational cost associated with using a metamodel is much lower than the standard approach of performing simulation runs for all configurations generated by the optimizer.

This paper presents a new population-based optimisation algorithm for SO that uses metamodels for efficiency enhancement, called Metamodel Enhanced (MME) Global Search. Although the basic motivation for MME Global Search is to be used in SO, it can be applied to any general optimisation scenario in which a computationally expensive evaluation function can be approximated by some fast metamodel. In the demonstration of the algorithm in this paper an Artificial Neural Network (ANN) based metamodel is used, however any adaptive metamodeling technique is possible. The intended application of the MME Global Search Algorithm is primarily on complex problems that are suitable to be solved with global search techniques, for example, those with multiple optima.

2 RELATED WORK

ANNs are mathematical models that attempt to imitate the behaviour of biological brains. ANNs have universal approximation characteristics and also the ability to adapt to changes through training. Instead of following a set of rules, they are able to learn underlying relationships between inputs and outputs from a collection of training examples and to generalize these relationships to previously unseen data. These attributes make ANN based metamodels very suitable to be used as the substitutes for computationally expensive simulation models.

In most ANN based simulation metamodelling approaches, after the training of the metamodels the simulation models are completely substitued during the optimisation process. These approaches can only be successful when there is a small discrepancy between the outputs from the metamodel and the simulation. Due to lack of data and the high complexity of real-world problems, it is generally difficult to develop a metamodel with sufficient approximation accuracy that is globally correct and metamodels often suffer from large approximation errors which may introduce false optima (Jin et al. 2002). One way to handle this problem and assure that the optimisation algorithm is not misled when the complexity of the fitness landscape is high is to alternate between the metamodel and the simulation model during the optimisation. Some of the work in combining simulation, ANN metamodels, and population-based optimisation are described in the rest of this section.

Bull (1999) presents an approach where an ANN metamodel is used in conjunction with a costly evaluation function to increase the efficiency of the optimisation. The neural network is first trained with a number of initial samples to approximate a theoretical model and a GA then uses the metamodel for fitness evaluations. For every 50 generations, the fittest individual in the population is evaluated using the original fitness function. This individual replaces the sample representing the lowest fitness in the training data set and the ANN is then retrained. The authors found that the GA is misled by the ANN when the fitness landscape of the modelled system is complex.

Jin et al. (2002) propose an approach for managing metamodels in population-based evolution. The main idea of this approach is that the frequency at which the original function is called and the metamodel is updated are determined by the estimated accuracy of the metamodel. The authors introduce the concept of evolution control and propose two control methods; controlled individuals and controlled generations. With controlled individuals, part of the individuals in a population are chosen and evaluated using the original fitness function. The controlled individuals can be chosen either randomly or according to their fitness values. With controlled generations, the whole population of *N* generations are evaluated with the original fitness function in *M* generations ($N \leq M$). On-line learning of the metamodel is applied after each call to the original fitness function when new training data are available. The authors carry out empirical studies to investigate the convergence properties of the implemented evolution strategy using an ANN-based metamodel on two benchmark problems. The authors found that incorrect convergence will occur if the metamodel has false optima.

Khu et al. (2004) discuss the integration between evolutionary algorithms and metamodels and propose a strategic and periodic scheme of updating the metamodel to ensure that the metamodel is constantly relevant as the search progresses. In the suggested approach, the whole population are first evaluated using the metamodel and the best individuals in the population are then evalutated using the true fitness function. The authors implement an ANN metamodel and a genetic algorithm for hydrological model calibration and show that there is a significant advantage in using metamodels for water and environmental system design.

Yan and Minsker (2004) propose a dynamic metamodelling approach, in which ANNs and support vector machines (SVM) are embedded into a GA. Data produced from early generations of the GA are sampled to train the ANN and SVM and the original evalution function is periodically called to dynamically update the ANN and SVM. The authors applied their proposed method to solve groundwater optimisation problems and results from their study show that satisfactory results can be achieved if the ANN metamodel is retrained or updated to fit the GA population in later generations.

Most approaches that make use of global search optimisation, simulations, and ANN-based metamodels employ the metamodel as an evaluation substitute in the ordinary evolutionary process. In contrast to previous studies, this paper presents an approach of using the metamodel for the probing of promising candidates to transfer to the next generation.

3 THE MME GLOBAL SEARCH ALGORITHM

The MME algorithm is a population-based metaheuristic optimisation algorithm. Figure 1 presents the general procedure of the algorithm. The algorithm maintains a population of solutions and progresses in increments called generations, where the population of each generation builds upon the previous generation. The basic principle of the algorithm is to generate a large number of candidate solutions and use a metamodel to choose the most promising ones to transfer to the next generation. There are two assumptions: (1) a good metamodel should be able to dismiss inferior solutions and thus avoid wasting valuable simulation time, and (2) the time required for computing the metamodel is negligible when compared to a simulation run. During the optimisation, the accuracy of the metamodel is constantly improved through applying on-line training after each generation of solutions have been simulated.

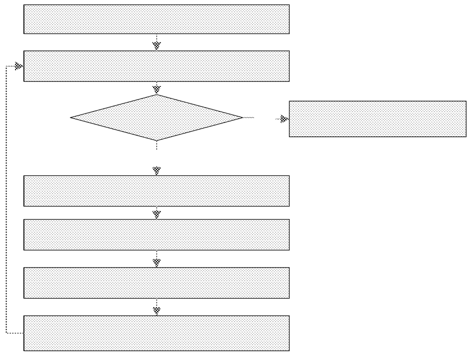


Figure 1: The MME Global Search Algorithm

3.1 Algorithm Core

A solution is defined by a triple $\left(input, mm\_output, sim\_output\right)$ where $input$ is an input sample, mm_output is the output produced by the metamodel, and sim_output is the output produced by the simulation. Any of these attributes can be unassigned. For example, if sim_output is unassigned, this means that the solution has not been simulated. In order to refer to the attributes of a solution, a subscript notation is used, e.g., i_{input}

is the *input* attribute of solution i. The algorithm core comprises three functions; MME_Global_Search, Evaluate_Population, and Evaluate_ Candidates. The main function is MME_Global_Search, which calls Evaluate_Population for evaluation of solutions in a population and Evaluate_ Candidates for evaluating candidate solutions in the population (Figure 2).

```
function MME_Global_Search( )
Returns: Best solutions found.
  $g \leftarrow 0$
  $population_0 \leftarrow$ Generate_Initial_Population( )
  Evaluate_Population($population_0$)
  Train_Meta_Model($\{population_0\}$)
  while (not Stop_Optimization( )) do
    $g \leftarrow g+1$
    $candidates_g \leftarrow$ Generate_Candidates($population_{g-1}$)
    Evaluate_Candidates($candidates_g$)
    $population_g \leftarrow$ Choose_Solutions($population_{g-1}, candidates_g$)
    Evaluate_Population($population_g$)
    Train_Meta_Model($\{population_0, \ldots, population_g\}$)
  end
  return Best_Solutions($\bigcup_{i=0}^{g} population_i$)

function Evaluate_Population(population)
  foreach p in population do
    if $p_{output}$ = null then
      $p_{output} \leftarrow$ Run_Simulation($p_{input}$)
    end
  end

function Evaluate_Candidates(candidates)
  foreach p in candidates do
    $p_{rough\_output} \leftarrow$ Run_Meta_Model($p_{input}$)
  end
```

Figure 2: Algorithm Core Functions

3.2 Problem-Specific Functions

The algorithm calls a number of problem-specific functions, which are described in this section.

function Generate_Initial_Population( )

Returns: A set of input samples.

Generates an initial population for the algorithm. One way to do this is to randomly generate a set of points in the search space. Alternatively, a more structured approach, such as Design of Experiments, can be used. In some cases, fairly good solutions already exist (e.g., from earlier optimisations) and can be directly returned from this function.

function Stop_Optimization( )

Returns: True to stop the optimisation.

This can, for example, be based on the quality of the solutions, time passed, or on the number of iterations since the best solution was found.

function Generate_Candidates(*population*)

Input: A population of solutions.

Returns: A set of candidate solutions.

Generates a set of candidate solutions from a population. The returned set of candidate solutions should normally be much larger than the population. The way that new solutions is generated is dependant on the solution input representation and may also include problem-specific heuristics. New solutions may, for example, be generated by using a combination of crossover and mutation operators (as used by a Genetic Algorithm). A specific implementation of this function is described in Section 3.3.

function Choose_Solutions(*previous_population*, *candidates*)

Input: The previous population and a new set of candidate solutions.

Returns: A set of solutions.

This function chooses the most promising solutions to use as the population for the next generation. This is normally based on the fitness of the solutions, but may also penalize similar solutions to keep the population diversified. The reason that the best candidates in the previous population is passed to the function is to support elitism, i.e. to keep some promising solutions for the new population. Through the conditional check in Evaluate_Population, solutions that are kept from the previous generation have no need to be simulated again.

function Best_Solutions(*solutions*)

Input: A set of solutions.

Returns: The best solutions.

Returns the best solutions based on user-defined criteria, such as the Pareto front.

function Run_Simulation(*input*)

Input: An input sample.

Returns: Output response from simulation.

Runs the accurate, but time-consuming, simulation.

function Train_Meta_Model(*populations*)

Input: A set of simulated populations.

Trains the metamodel with the given solutions or a subset thereof.

function Run_Meta_Model(*input*)

Input: An input sample.

Returns: Output response from metamodel.

This function is assumed to be many orders of magnitude faster than Run_Simulation. Note that the structure of the output returned from this function may not be the same as returned from Run_Simulation. The metamodel may, for example, return an objective value as output instead of the output type returned by the simulation.

3.3 Specialisation of Algorithm

This section describes a specific implementation of the function Generate_Candidates based on the concepts of Genetic Algorithms (Figure 3). The function contains two constants, *num_candidates* and *crossover_frequency*. The constant *num_candidates* specifies the number of candidate solutions to generate and it is assumed to be an even number. Three problem-specific functions are used, which are explained below.

function Select(*population*)

Input: A set of solutions.

Returns: One of the solutions from the set.

Selects a solutions based on its fitness in relation to the other solutions. A number of popular selection schemes exists, such as roulette wheel, ranking, and tournament.

function Crossover$(solution1, solution2)$

Input: Two parent solutions.
Returns: Two child solutions.

Generates two new solutions based on a recombination of the two parent solutions.

function Mutate$(solution)$

Input: A solution.
Returns: A mutated solution.

Returns a solution that is a mutated version of the given solution.

```
function Generate_Candidates($(population)$)
Returns: Returns $num\_candidates$ candidates.
  $candidates \leftarrow \varnothing$
  for $i \leftarrow 1$ to $\frac{num\_candidates}{2}$ do
    $a \leftarrow$ Select$(population)$
    $b \leftarrow$ Select$(population)$
    with probability $crossover\_frequency$ do
      $(c,d) \leftarrow$ Crossover$(a,b)$
    else
      $(c,d) \leftarrow (a,b)$
    end
    $candidates \leftarrow candidates \cup \{$Mutate$(c)$, Mutate$(d)\}$
  end
  return $candidates$
```

Figure 3: Variant of Algorithm

4 BENCHMARK TEST PROBLEM

This section demonstrate the MME Global Search algorithm applied to the 2-D Rosenbrock optimisation benchmark problem (Equation 1). This is used as an initial benchmark function by assuming that it represents a computationally expensive evalution function.

$$f(x,y) = 100\left(y - x^2\right)^2 + (1 - x)^2 \qquad (1)$$

The fitness landscape of the Rosenbrock function (plotted in Figure 4) has a global minimum of 0 at the point (1, 1).

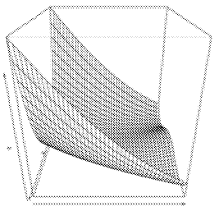


Figure 4: The 2-D Rosenbrock Function

An ANN is developed as a fast metamodel of the Rosenbrock function. It is trained to estimate the Rosenbrock function as a function of a search space coordinate. A feed-forward network with two hidden layers is constructed with two input nodes, 20 nodes in each hidden layer and one output node. While a single hidden layer may be sufficient to approximate any continuous function, it is not always optimal in terms of learning time (Chester 1990). One hidden layer may require an infinite number of neurons to approximate a given function and the use of two hidden layers can avoid this assumption. The data set which is used to train the ANN consists of 1000 input-output pairs randomly generated in the search space.

4.1 Implementation

An input sample consists of a vector of 2 real values, corresponding to a X-Y coordinate in the search space. An output sample consists of a real value, corresponding to the Rosenbrock function value. The goal of the optimisation is to minimise this value. The implementation uses the GA-based variant of Generate_Candidates described in Section 4.3. A population size of 10 indivudals is used and the initial population is randomly initated. In each iteration of the algorithm, 100 candidate solutions are generated and evaluated using the ANN. Solutions are selected for reproduction using tournament selection in which two solutions are randomly chosen and the one with the lowest objective function value is returned. A standard one-point crossover is used and recombination of solutions occurs with a probability of 0.5. Each value in the solutions is mutated using a Gaussian distribution with a deviation that is randomly selected from the interval [0,10] for each individual. During the search, the ANN is trained continuously with incremental training using back-propagation for 1 epochs with a learning rate of 0.1. Since the training set in this test is large, only data from the last evaluated population is used to train the network and the training continues from the last weights. The optimisation runs for 100 iterations before it terminates and returns the best solution found.

4.2 Results

This section presents the results of the MME Global Search implementation described in the previous section. For comparison, a standard GA is implemented for the same optimisation problem and simulation model. This algorithm uses the same representation, objective function, crossover operator and mutation operator as the MME Global Search implementation.

In Figure 5, average results from 5000 replications of the two experiments are shown. The chart shows the best fitness value found against the number of function evaluations.

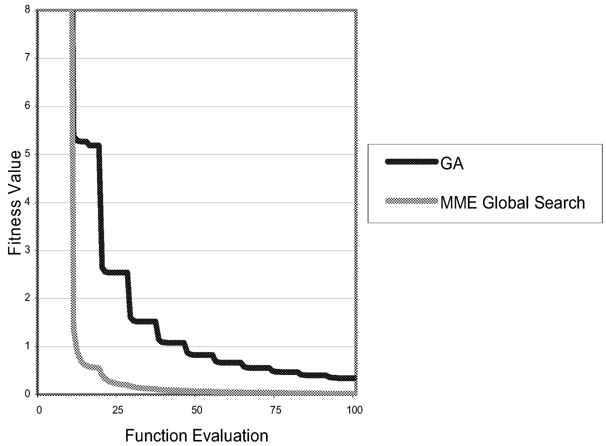


Figure 5: Comparison of Experiments

As the chart shows the MME Global Search algorithm converge faster than the standard GA.

Using on-line training, the accuracy of the metamodel is continuously improved, as shown in Figure 6. This figure presents the Mean Square Error (MSE) of the metamodel estimated locally based on the information from the last simulation. The MSE presented is an average of 5000 replications.

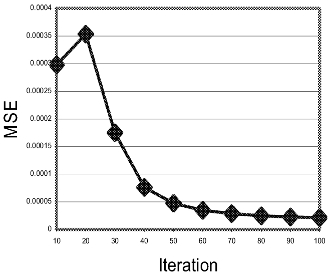


Figure 6: Estimated Metamodel MSE

5. CONCLUSIONS AND FUTURE WORK

This paper presents a new population-based meta-heuristic algorithm for simulation-based optimisation. The proposed algorithm uses a metamodel for efficiency enhancement. Similar to other population-based metaheuristics, such as Genetic Algorithms (GA), it generates and maintains a large population of solutions and progresses incrementally generation by generation using genetic operations. The difference is that it generates a large set of candidate solutions followed by the use of a metamodel to probe for the most promising ones. During the optimisation, the accuracy of the metamodel is constantly improved as on-line training is applied after every individual in a generation of solutions has been simulated.

The proposed algorithm is implemented on a benchmark optimisation problem and results show that the efficiency of the optimisation process could be significantly enhanced by avoiding running time-consuming simulations for low-quality candidate solutions. Initial results also indicate that the proposed algorithm shows good performance in comparison with a standard GA-based approach.

Future work will focus on many different aspects of this ongoing research. These include verification of the proposed algorithm by applying it to different real-world optimisation problems with various properties, and performing further benchmarkings with some other state-of-the-art algorithms. Planned future work also includes combining the algorithm with the metamodel-based local search algorithm presented in Persson et al. (2006).

An intresting variant of the algorithm is to choose the *n* best candidate solutions to transfer to the next generation not exclusively based on fitness values. When there is a large number of candidate solutions, there is a risk that a great proportion of the generated candidates are identical to one another and as a consequence the new population becomes more or less homogeneous. To prevent this, a diversification control can be implemented, punishing candidate solutions that are too close to each other. Planned future work will include the testing of this variant of the proposed algorithm.

When using ANN and population-based search methods a large number of variables exist, such as number of hidden layers, number of nodes in the hidden layers, population size, mutation rate, etc. In future work, the effects of varying these parameters will be tested. Further, for ANNs the data samples in the intial training set can have large influence on the networks' approximation ability. Ways to achieve more effective training of the ANN will be investigated in the future.

The higher the accuracy of the metamodel, the more frequently it can be used and hence the time consumption of the optimisation process can be reduced. However, it is very difficult to estimate the global accuracy of the metamodel and future work includes investigating how one can get an understanding of the metamodels influence on the search direction. Currently the whole population of solutions is evaluated using the simulation model, but with an adequate metamodel it may, for example, be sufficient to evaluate only a subset of the population with the simulation and using the metamodel for evaluating the other part.

REFERENCES

Alam, F., K.R. McNaught; and T.J. Ringrose. 2004. "A comparison of experimental designs in the development of a neural network simulation metamodel". *Simulation Modelling Practice and Theory*, Vol.12 No.7-8, 559–578.

April, J, M. Better, F. Glover; and J. Kelly. 2004. "New advances for marrying simulation and optimization". In *Proceedings of the 2004 Winter Simulation Conference* (Washington, D.C., Dec.5-8), 80-86.

Buchholz, P. and A. Thümmler. 2005. "Enhancing evolutionary algorithms with statistical selection procedures for simulation optimization". In *Proceedings of the 2005 Winter Simulation Conference* (Orlando, FL, Dec.4-7), 842-852.

Bull, L. 1999. "On model-based evolutionary computation". *Software Computing*, Vol.3, 76-82.

Chester, D. 1990. "Why two hidden layers are better than one". In *Proceedings of the International Joint Conference on ANNs* (San Diego, June 17-21), 265-268.

Jin, Y., M. Olhofer; and B. Sendhoof. 2002. "A Framework for Evolutionary Optimization With Approximate Fitness Functions". *IEEE Transactions on Evolutionary Computation*, Vol.6 No.5 (Oct), 481-494.

Khu, S.T., D. Savic, Y. Liu; and H. Madsen. 2004. "A fast Evolutionary-based Metamodelling Approach for the Calibration of a Rainfall-Runoff Model". In *Proceedings of the First Biennial Meeting of the International Environmental Modelling and Software Society* (Osnabruck, Germany), 147-152.

Persson, A., Grimm, H. and Ng, A. 2006. "Simulation-Based Optimisation Using Local Search and Neural Network Metamodels". *In Proceedings of the 10th IASTED International Conference on Artificial Intelligence and Soft Computing* (Mallorca, Spain), 178-183.

Yan, S. and B. Minsker. 2004. "A Dynamic Metamodel Approach to Genetic Algorithm Solution of a Risk-Based Groundwater Remediation Design Model". In *Proceedings of World Water and Environmental Resources Congress* (Salt Lake City, Utah, June 27-July 1), 1-10.

OPTIMIZATION BY EXTENSION-RESTRICTION NEIGHBORHOOD IN LOCAL SEARCH: APPLICATION TO GRAPH COLORING PROBLEM

Isabelle Devarenne
Hakim Mabed
Alexandre Caminada

UTBM, University of Technology Belfort-Montbéliard, SET Lab,
90010 Belfort cedex, France
Emails: {isabelle.devarenne|hakim.mabed|alexandre.caminada}@utbm.fr

KEYWORDS
Local search, neighborhood extension/restriction, loop detection, graph coloring.

ABSTRACT

The efficiency of a search method depends on the exploration/intensification equilibrium. We present here a local search method based on intelligent extension and restriction neighborhood mechanisms defined by a loop detection procedure and a tabu list structure. To assess the performance of the method, the *K*-Coloring problem is chosen as a test application. Obtained results have shown very promising performances.

INTRODUCTION

In the literature, there exist two kinds of local search method according to the nature of neighborhood exploration. The first approach is based on complete neighborhood exploration method. An example of such algorithm is the basic Tabu Search presented in (Dorne and Hao 1998). The second approach is based on partial neighborhood exploration as in the works of (Hansen and Mladenovic 2003) and (Lourenco et al. 2002).

In this paper, an adaptive local search method is presented. This method is based on partial neighborhood exploration. Loop detection and tabu list structure are used as mechanisms serving to intelligently exploring the neighborhood. The term loop refers to the move choice redundancy and not the fact that identical sequences of moves are observed. Loop detection and tabu list structure are adaptively used, in order to obtain more suitable behavior of the search process. This is attained by a more judicious alternation of extension and restriction phases of the search.

This approach is applied on k-coloring problem, one of the most studied optimization problems. The method is applied and evaluated on standard benchmark like DIMACS. The objective of the *K*-coloring problem is to color the nodes of a given graph with *K* colors. To do so, the following constraint is to be respected: different colors must be assigned to any couple of neighbor nodes. There exist many practical applications in resource management for *K*-coloring model. Heuristic methods are used to deal with the *K*-coloring, knowing the NP-completeness of such problems. Therefore several heuristic methods were proposed such as: constructive methods (DSATUR (Brelaz 1979)), tabu search (Dorne and Hao 1998), reactive search (Battiti 2005) and evolutionary approach (Galinier and Hao 1999) (Galinier et al. 2004). Two new local search methods based on adaptive extension and restriction of neighborhood have been recently proposed. These methods are Variable Neighborhood Search (VNS) (Hansen and Mladenovic 2003) and Iterated Local Search (ILS) (Chiarandini and Stutzle 2002) (Lourenco et al. 2002) (Chiarandini 2005) .

After the introduction, a formal description of the *K*-coloring is given. In the third and fourth sections, loop detection and tabu list mechanisms are described. In the fifth section, an analysis of numerical results on DIMACS problems is presented and compared to the best published. And finally, a general conclusion and perspectives are presented.

BASIC LOCAL SEARCH METHOD APPLIED TO *K*-COLORING PROBLEM

For a given undirected graph $G = (V, E)$ with $V = \{v_1,...,v_N\}$ the set of vertices and $E=\{e_{ij} \mid \exists \text{ an edge between } v_i \text{ and } v_j\}$ the set of edges, the k-coloring problem goal is the minimization of the conflicts number, using only K available colors. A conflict is produced when two adjacent nodes use the same color as described in the following expression: $\forall e_{ij} \in E,\ c_i \neq c_j$ where, $c_i \in [1..K]$ is the color assigned to the v_i node. A solution s of the problem is represented by a vector of colors $<c_1,...,c_N>$. Based on this representation, the combinatorial search space size is equal to K^N.

In order to reach the best solution, local search method performs iterative moves on a single solution. In the proposed method, fitness degradation control is not used. This implies that all the selected moves are taken whatever the gain or loss brought by the neighbor solution. Because there is no anticipation on the future, the iteration execution is very fast. From the other hand, the choice problem of a degradation parameter value is eliminated.

A move from a solution *s* to *s'* is performed at each iteration. Every move changes only one node color. Two solutions *s* and *s'* are defined as neighbors if and only if they differ by a single node color. To each solution *s*, the fitness function *F* (F: $S \rightarrow \mathbb{N}$) associates an integer value *F(s)* which refers to the unsatisfied constraints number, which is equal to zero if *s* satisfies all the constraints.

Node selection is done at each iteration, according to one of the three following methods: (a) randomly among the most conflicted nodes; (b) randomly among the conflicting nodes, whatever their conflicts number is, or (c) randomly among all nodes. Method (a) is the most deterministic method, compared to methods (b) and (c), (c) being the less deterministic one. Each node selection method can be considered as a specific operator that uses different degrees of neighborhood diversity. After node selection, a new different color is chosen and assigned to it according to two methods. Color method (a) chooses a color randomly among all existing colors; whereas, in (b), a color is chosen randomly among the less used colors of the node's neighborhood. The efficiency of the use of the node choice (a) in combination with the color choice (b) has been proved by the study performed in (Devarenne et al. 2005). In this work, the relevance of adaptive (intelligent) neighborhood exploration combining several move operators is studied. To this end, the loop detection mechanism was introduced and at second time the tabu list structure is described in details in the following sections.

LOOP DETECTION MECHANISM AND ADAPTIVE OPERATORS COMBINATION

A novel diversification procedure based on loops detection is proposed, in addition to these basic mechanisms already studied in literature. This procedure is also referred in (Devarenne et al. 2005). The node selection based on the method (a) is very deterministic, increasing the risk of stuck in local optimum. Extending the neighborhood choice is used to improve the search exploration. A node causes a loop when it is frequently selected during the *n* (*n=N/2*) last iterations. In this case, the deterministic choice of the node is replaced by a diversifying method during the next iteration. Deterministic iteration consists on choosing the most conflicted node and assigning to it the less used color. In diversifying iteration, the node is randomly chosen among all conflicted nodes. For both iterations types, color choice remains the same. In Fig. 1 the working scheme of the local search method is presented.

In order to understand the interaction between deterministic and diversifying mechanisms during the search, Fig. 2 presents the number of consecutive diversifying iterations between two deterministic ones. In this figure, the most difficult CNET instance (Galinier and Hao 1999) named *8.150.20* is used, where, *8* is the chromatic number, *150* the number of nodes and *20* the graph density (ratio between the edges number in the graph and the number of nodes pairs). Only the last 100 iterations before reaching a global optimum of a search are plotted in this curve. Between two deterministic choices, the consecutive diversifying iterations vary from 0 to 6. At the end of the run, the number of conflicted nodes decreases causing more loops detection and therefore more diversifying iterations are applied.

```
DeterministicIteration()
Begin
   Select the most conflicted node
   Select the less used color in the neighborhood
End

DiversifyingIteration()
Begin
   Select a conflicted node
   Select the less used color in the neighborhood
End

Local Search Algorithm
Begin
   Generate the initial solution
   DeterministicIteration()
   While nbIter<nbIterMax and not solOptimFound
      If loopDetection() then
         DiversifyingIteration()
      Else DeterministicIteration()
   End while
End
```

Figure 1: Loop detection mechanism algorithm

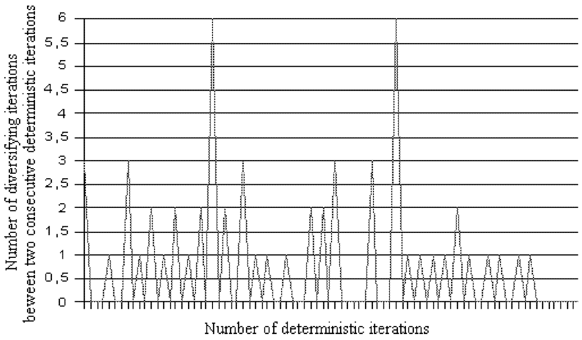


Figure 2: Diversifying iterations between two deterministic

The loop detection mechanism allows the diversification of the selected nodes when the choices become more restrictive, and therefore improve the search quality. After the next iteration, nodes causing loops can be chosen again because that loop detection mechanism does not make any restriction on the selection of theses nodes. To orient search and avoid repetition of nodes choices, a tabu list is used.

TABU LIST MECHANISM

An important component of Tabu Search (TS) algorithm (Glover 1989) is the *tabu list*. The main idea of TS is to choose the best move at each iteration. So it is a huge deterministic process. To avoid cycles and repetitions, the method memorizes the selected moves or the visited solutions inside a list of non-return called *tabu list*. Then, this short-term memory is used to *restrict* the neighborhood of any current solution to a subset of admissible neighbors. The main parameter of the tabu list is the tabu tenure that defines the number of next iterations during which the solution or the move still forbidden. In literature, the tabu tenure is often defined dynamically (Dorne and Hao 1998).

We introduce the tabu list mechanism in addition to loop detection to memorize the nodes detected inside the loops and to give them a tabu status for a given duration. For that, we checked two kinds of tabu list: a list of tabu nodes, i.e. the selection of these nodes is forbidden for a given number of iterations and a list of tabu assignments (node, color) where a given color is forbidden for a given node. Firstly, we used fixed tabu tenure, i.e. the value is the same for all nodes during all the search. Then, we used dynamic tabu tenure, i.e. for each node the tabu tenure is randomly chosen in $[0.5\times f(N)\ ;\ 1.5\times f(N)]$ where *f(N)* is a function of the number of nodes in the graph.

The context in which we are using the tabu list is very different from the Tabu Search. In our local search method we are not exploring the neighbors of the current solution and we are not selecting the next solution looking at its performances. When the move is selected the new solution becomes the current even with worst quality.

Now the problem is to compare all these parameters settings. The next section presents a succession of experiments to measure interactions between loop detection and tabu nodes.

EXPERIMENTAL RESULTS

The aim of this section is to discuss several results on the combination of loop detection mechanism and tabu list in our local search algorithm. Our analysis has been done on DIMACS instances which are commonly used in research papers (Chiarandini 2005) (Dorne and Hao 1998) on *k*-coloring problem.

Table 1 exposes the results obtained for Leighton graphs. For all instances, the chromatic number is known and it is given in the second column. We compare five methods. The first algorithm is based on loop detection mechanism without tabu list. Four others algorithms are analyzed, all implementing loop detection and tabu list mechanisms. The second and third methods use a static tabu tenure equal to $\sqrt{N}/2$, whereas the fourth and the fifth algorithms use dynamic tabu tenure inside $[0.5\times\sqrt{N}/2;\ 1.5\times\sqrt{N}/2]$. In the second and the fourth algorithms, the tabu list only records nodes appearing in loops, whereas the two others methods record all visited nodes. Two comparison criteria are used to perform this analysis: the success rate over 10 runs, and the average number of iterations needed to obtain the optimal solution.

The introduction of the tabu list to durably eliminate nodes from selection greatly increases the success rate of the method. Results of static and dynamic tabu tenure parameters are really variable in term of success rate and average number of iterations.

At first sight we may conclude that, when all selected nodes become tabu the local search always or never find the optimal solution on the problems whatever the static or dynamic tabu tenure. On the opposite, when the tabu status is assigned to nodes detected inside loops, it seems that the method is not so binary; for example, with the static tabu tenure 10 problems on 12 were solved optimally but not with *100%* of success. Table 1 presents some particularities for problems *le450_15a, le450_15b, le450_15c* and *le450_15d*. For these problems, the results differ according to the selected nodes for tabu status, i.e. at each iteration or when loops are detected. For *le450_15a*, the methods 3 and 5 never found an optimal solution while the two others methods did it. Inverse occurs for instances *le450_15c* and *le450_15d* where we move from *100%* to *0%* or *10%* of success.

After these first tests, we checked the influence of tabu tenure duration on the process. We know that this parameter has a great influence on TS method and we must check it on our method too. We did it on dynamic tabu tenure which is never able to find one optimal solution for problems *le450_15a* and

Table 1: Influence of tabu list on Leighton graphs

		Loop detection mechanism									
		without tabu list		tabu list with static tabu tenure				tabu list with dynamic tabu tenure			
				after loop		every iteration		after loop		every iteration	
DIMACS	k	succ	nb eval	succ	nb eval	succ	nb eval	succ	nb eval	succ	nb eval
le450_5a	5	70%	377528	50%	234351	**100%**	9679	30%	302000	**100%**	7903
le450_5b	5	40%	169875	10%	125511	**100%**	26870	10%	629000	**100%**	28870
le450_5c	5	90%	207041	80%	460771	**100%**	9070	**100%**	252120	**100%**	6107
le450_5d	5	80%	102871	90%	230401	**100%**	3845	80%	396000	**100%**	5479
le450_15a	15	40%	387305	90%	452579	0%		50%	304000	0%	
le450_15b	15	80%	44108	80%	279306	0%		90%	390000	0%	
le450_15c	15	0%		10%	235273	**100%**	183202	0%		**100%**	194312
le450_15d	15	0%		10%	128381	**100%**	294031	0%		**100%**	184193
le450_25a	25	90%	2052	**100%**	2336	**100%**	368317	90%	2526	**100%**	179095
le450_25b	25	**100%**	1109	**100%**	1061	**100%**	6109	**100%**	1198	**100%**	8595
le450_25c	25	0%		0%		0%		0%		0%	
le450_25d	25	0%		0%		0%		0%		0%	

Table 2: Influence of the tabu tenure values on the algorithm performances

		Loop detection mechanism with every visited nodes is tabu for a dynamic tabu tenure														
		$\sqrt{N}/8$			$\sqrt{N}/7$			$\sqrt{N}/6$			$\sqrt{N}/5$			$\sqrt{N}/4$		
DIMACS	k	succ	nb eval	c	succ	nb eval	c	succ	nb eval	c	succ	nb eval	c	succ	nb eval	c
le450_15a	15	**100%**	42911		0%		4	50%	147775	2	0%		2	0%		4
le450_15b	15	0%	8	1	0%		4	50%	296529	2	0%		1	0%		4

le450_15b. We decided to progressively reduce the interval of variations of dynamic tabu tenure; doing that the method becomes more and more static-like. The results are presented in table 2 with 5 different values for *f(N)*. Three criteria are now observed: success rate named *succ*, average number of iterations named *nb_eval* and also, *c*, which is the average number of conflicts of the best solutions when no optimal solution was found. These results confirm the importance of the tabu tenure in our method. The problems are solved for some parameters and sometime with *100%* of success. The previous value *f(N)*= $\sqrt{N}/2$ was too high and therefore too restrictive for *le450_15a* and *le450_15b* instances; smaller values of *f(N)* give better results.

Obviously, this value should be adapted according to the content of the tabu list (nodes list or assignments list) and to the frequency of the tabu status (after loop detection or after each iteration). So we cannot definitively conclude on the best combination. Automatic parameters settings of tabu tenure will be proposed as further works.

As last work on parameters settings, table 3 presents results for random graphs DSJC (still DIMACS instances). The chromatic number is only known for the first instance. Then for the others, we used the best result known in literature (second column). We only present 4 methods. The first one refers to the deterministic method without loop detection and tabu list. Other methods use loop detection mechanism: without tabu list, with a tabu list based on loops nodes and finally, with a tabu list based on all selected nodes. All methods use dynamic tabu tenure with *f(N)*= $\sqrt{N}/2$. From this table, we see that combination of loop detection and tabu list mechanisms gives the best results when loop nodes become tabu.

COMPARATIVES RESULTS

The objective of this section is to compare our current results with already published ones. Therefore, we compare our method with four algorithms on random graphs DSJ (see table 4). We use the results published in (Galinier et al., 2004), in (Dorne and Hao, 1998) and in (Chiarandini et al., 2005). The results presented in table 4 concern four methods: DSATUR, AMACOL methods described in (Galinier et al., 2004), ILS described in (Chiarandini et al., 2005), and finally Generic Tabu Search described in (Dorne and Hao, 1998).

- AMACOL (Adaptive Memory Algorithm for *K*-COLoring) is a population-based method and represents one of the more efficient algorithms.
- ILS (Iterated Local Search) method that uses a permutation of the color classes as a perturbation operator (Chiarandini et al. 2005). A color class is a set of nodes having the same color.
- GTS (Generic Tabu Search) published by Dorne and Hao (Dorne and Hao, 1998) is based on a Tabu Search approach. This method uses dynamic tabu tenure and a greedy initial solution made by DSATUR. The tabu tenure depends on the number of conflicted nodes and a stochastic number.
- Finally, we present our results in the last column from the combination of loop detection (LD) and tabu list (TL) on loops' nodes with dynamic tabu tenure. For each method, we give the minimum number of colors used to resolve all conflicts for each problem instance.

All random graphs DSJC are classified as hard instances (Chiarandini 2005). AMACOL solves all instances with the minimum number of colors. DSATUR method never obtains

Table 3: Influence of tabu list on random graphs DSJC

		used method											
		without loop detection			loop detection mechanism								
		without tabu list			without tabu list			after loop			every iteration		
DIMACS	k	succ	nb eval	c	succ	nb eval	c	succ	nb eval	c	succ	nb eval	c
DSJC125.1	**5**	**100%**	27959		**100%**	24294		**100%**	26421		**100%**	44395	
DSJC125.5	17	**100%**	31.8e+06		**100%**	1.7e+6		**100%**	1.7e+06		80%	31.1e+06	1
DSJC125.9	44	**100%**	23742		**100%**	18116		**100%**	22109		80%	21801	1
DSJC250.1	8	**100%**	3.4e+06		**100%**	201656		**100%**	66514		**100%**	2.2e+06	
DSJC250.5	28	0%		4	0%		2	**100%**	29.6e+06		0%		4
DSJC250.9	72	0%		1	30%	172870	1	50%	5.2e+06		0%		1
DSJC500.1	12	0%		7	0%		9	0%		2	0%		5
DSJC500.5	48	0%		17	0%		16	0%		5	0%		16
DSJC500.9	126	0%		17	0%		7	0%		5	0%		17
DSJC1000.1	20	0%		16	0%		26	0%		15	0%		16
DSJC1000.5	84	0%		39	0%		63	0%		46	0%		39
DSJC1000.9	224	0%		--	0%		--	0%		7	0%		34

Table 4: Comparatives results on DSJC instances

DIMACS	DS	AMACOL	ILS	GTS	LD+TL
DSJC125.1	6	**5**	**5**	**5**	**5**
DSJC125.5	21	**17**	**17**	**17**	**17**
DSJC125.9	50	**44**	**44**	**44**	**44**
DSJC250.1	10	**8**	**8**	**8**	**8**
DSJC250.5	38	**28**	**28**	**28**	**28**
DSJC250.9	91	**72**	**72**	**72**	**72**
DSJC500.1	16	**12**	13	13	13
DSJC500.5	67	**48**	50	50	50
DSJC500.9	161	**126**	127	127	128
DSJC1000.1	26	**20**	21	21	21
DSJC1000.5	114	**84**	90	90	89
DSJC1000.9	297	**224**	227	226	230

the best results. Both ILS and GTS methods allow obtaining an optimal solution for the six first instances. Our current method globally performs as GTS on solved and unsolved instances. We must check if other parameters are more efficient for this particular set of instances.

CONCLUSIONS AND PERSPECTIVES

In this paper, we introduced a new local search method based on partial exploration of solution neighborhood. The neighborhood exploration uses two complementary mechanisms: loop detection and tabu list procedures. Loop detection mechanism leads to the extension of neighbor choice, whereas tabu list mechanism aims at restricting this choice to avoid the combinations already explored.

To assess the performance of the proposed method we tested it compared to four other algorithms presented in literature. The results showed the great competitiveness of our method on some hard instances of DIMACS benchmark.

However, the method appears very sensitive to parameter settings. As future work, we will study the self-tuning of the tabu tenure and the loop detection parameter α.

REFERENCES

Battiti R. and M. Brunato. 2005. “Reactive search: Machine Learning for memory based heuristics”, Technical Report #DIT-05-058, Università Degli Studi di Trento, Trento, Italy (Sept).

Brelaz D. 1979. “New Methods to Color the Vertices of a Graph”. *Communications of the ACM* 22, No. 4,251-256.

Chiarandini M. and T. Stützle. 2002. “An application of Iterated Local Search to Graph Coloring”. In *Proceedings of the Computational Symposium on Graph Coloring and its Generalizations*, D. S. Johnson, A. Mehrotra and M. Trick (Eds), 112-125.

Chiarandini, M.; I. Dumitrescu; and T. Stutzle. 2003. “Local Search for the Colouring Graph Problem, A Computational Study”. Technical Report AIDA-03-01.

Chiarandini, M.; I. Dumitrescu; and T. Stutzle. 2005. “Stochastic Local Search Algorithms for the Graph Colouring Problem”. Technical Report AIDA-05-03.

Chiarandini M. 2005. “Stochastic Local Search Methods for Highly Constrained Combinatorial Optimisation Problem”. Ph.D. thesis, Computer Science Department, Darmstadt University of Technology, Darmstadt, Germany.

Devarenne I.; H. Mabed; and A. Caminada. 2005. “Analysis of adaptive local search for graph coloring problem”. In *MIC'2005, 6th Metaheuristics International Conference*, 272-277.

Dorne R. and J.K. Hao. 1998. “Tabu Search for graph coloring, T-coloring and Set T-colorings”. In *Metaheuristics 98: Theory and Applications*, I.H. Osman et al. (Eds.), Kluver Academic Publishers.

Galinier P. and J-K. Hao. 1999. “Hybrid Evolutionary Algorithms for Graph Coloring”. In *Journal of Combinatorial Optimization* 3, 4, 379-397.

Galinier P. and A. Hertz. 2004. “A survey of local search methods for graph coloring”. Technical Report Les Cahiers du GERAD G-2004-37, GERAD and École des Hautes Études Commerciales.

Galinier, P.; A. Hertz; and N. Zufferey. 2004. “An Adaptive Memory Algorithm for the k-colouring problem”. Les Cahiers du GERAD G-2003-35.

Glover F. 1989. “Tabu search - part I”. *ORSA Journal on Computing* 1, No. 3, 190-206.

Hansen P. and N. Mladenović. 2003. “Variable neighborhood search”. In *Hanbook of Metaheuristics*, Glover and Kochenberger (Eds.), 145-184, Kluwer Academic Publisher.

Lourenco H.; O. Martin; and T. Stützle. 2002. “Iterated Local Search”. In *Handbook of Metaheuristics*, F. Glover and G. Kochenberger (Eds.), 321-353, Kluwer Academic Publishers, Norwell, MA.

EXPERIMENTAL-BASED MODELING AND PARETO OPTIMIZATION OF INDIRECT INJECTION DIESEL ENGINES

K. Atashkari, N. Nariman-zadeh, A. Jamali
Department of Mechanical Engineering,
Engineering Faculty, University of Guilan
P.O. Box 3756, Rasht, Iran
E-mail:atashkar@guilan.ac.ir

Ì. Çelikten
Technical Education Faculty, Gazi University,
Teknikokullar, 06503 Ankara, Turkey,
E-mail: celikten@gazi.edu.tr

KEYWORDS
Injection pressure, Diesel engine, GMDH, Genetic Algorithms, Multi-objective

ABSTRACT

Variable injection-pressure can be used to reduce emissions and increase engine Power. Experiments have been performed on a turbocharger diesel engine for four different injection pressures at three engine throttle positions. The measurements values of the engine Power and NO_x emission have been investigated. In this investigation, group method of data handling (GMDH)-type neural network and evolutionary algorithms (EAs) are used for modeling of the effects of the engine speed (N), throttle-position (TP) and injection-pressure (IP) on both engine Power and NO_x emission using data provided in the experiments. Employing the obtained polynomial models, multi-objective EAs are then used for Pareto-based optimization of the engine considering two conflicting objectives (engine Power and NO_x).

INTRODUCTION

With the strengthening needs of automobile pollutant legislation and the continuous improvement of thermal efficiency of internal combustion engines, more comprehensive and detailed research has to be conducted. Earlier design objectives were driven mainly by efficiency, reliability and durability. Since low toxic emissions dominate engine design practice, reducing the level of emissions requires experimental and theoretical knowledge of combustion in engines (US-EPA. 1982; Geist 1996; Parlak et al. 2003 and Winterborne et al. 1994).

Of all internal combustion engines, the diesel engine is the most efficient – that is, it can extract the greatest amount of mechanical energy from a given amount of fuel. It achieves this high level of performance by compressing air to high pressures before injecting very small droplets of fuel into the combustion chamber and, consequently, has been widely used in transportation and electric generating plants. Thus, investigations on the performance of diesel engines have become one of the important subjects in the optimal design.

Diesel engines produce lower amounts of HC, CO and NO_x than comparable gasoline engines. HC and CO is lower because of more complete combustion of the fuel and air mixture. NO_x is lower because peak temperature is not maintained very long. Smoke, or particulate emissions, occurs when there is insufficient air to completely burn the fuel. There are several factors that the engine designer varies to provide low emission levels with high performance and good fuel economy. Some of these factors are the shape of the combustion chamber, the location and angle of the fuel nozzle, the injection rate and nozzle spray pattern, injection timing (Yang et al. 1996; Brneaux 2001 and Parlak et al. 2005). In present diesel engines, fuel injection systems and injection pressures are adjusted to achieve the spray at higher pressures. This is done to decrease the exhaust emissions which occur as a result of the combustion process (Çelikten 2003). Experimental investigations to measure the performance of, and emissions from, a diesel engine are complex, time consuming, and costly. To predict the emissions from the engines, one approach is using mathematical models. However, their accuracies may not be sufficiently high. The alternative to a mathematical model is the experiment-based approach.

System identification and modelling of complex processes using input-output data have always attracted many research efforts. In fact, system identification techniques are applied in many fields in order to model and predict the behaviours of unknown and/or very complex systems based on given input-output data (Astrom and Eykhoff 1971). Group Method of Data Handling (GMDH) algorithm is a self-organizing approach by which gradually complicated models are generated based on the evaluation of their performances on a set of multi-input-single-output data pairs (i=1, 2, …, M). The GMDH was first developed by Ivakhnenko (Mueller and Lemke 200) as a multivariate analysis method for complex systems modelling and identification. GMDH can be used to model complex systems without having specific knowledge of the systems. The main idea of GMDH is to build an analytical function in a feedforward network based on a quadratic node transfer function whose coefficients are obtained using regression technique.

Optimization in engineering design has always been of great importance and interest particularly in solving complex real-world design problems. Basically, the optimization process is defined as to find a set of values for a vector of design variables so that it leads to an optimum value of an objective or cost function. There are many calculus-based methods including gradient approaches to single objective optimization and are well documented in (Arora 1989). However, some basic difficulties in the gradient methods, such as their strong dependence on the initial guess, cause them to find local optima rather than global ones. Consequently, some other heuristic optimization methods, more importantly Genetic Algorithms (GAs) have been used extensively during the last decade. Such nature-inspired evolutionary algorithms differ from other traditional calculus based techniques. The main difference is that GAs work with a population of candidate solutions not a single point in search space. This helps significantly to avoid being trapped in local optima as long as the diversity of the population is well preserved. Such an advantage of evolutionary algorithms is very fruitful to solve many real-world optimal design or decision making problems which are indeed multi-objective. In these problems, there are several objective or cost functions (a vector of objectives) to be optimized (minimized or maximized) simultaneously. These objectives often conflict with each other so that improving one of them will deteriorate another. Therefore, there is no single optimal solution as the best with respect to all the objective functions. Instead, there is a set of optimal solutions, known as Pareto optimal solutions or Pareto front (Srinivas and Deb 1994) for multi-objective optimization problems. The concept of Pareto front or set of optimal solutions in the space of objective functions in multi-objective optimization problems (MOPs) stands for a set of solutions that are non-dominated to each other but are superior to the rest of solutions in the search space. This means that it is not possible to find a single solution to be superior to all other solutions with respect to all objectives so that changing the vector of design variables in such a Pareto front consisting of these non-dominated solutions could not lead to the improvement of all objectives simultaneously. Consequently, such a change will lead to deteriorating of at least one objective. The inherent parallelism in evolutionary algorithms makes them suitable for solving MOPs. The early use of evolutionary search is first reported in 1960s by Rosenberg (Rosenberg 1967). Since then, there has been a growing interest in devising different evolutionary algorithms for MOPs.

In this study, GMDH-type neural networks are used for modelling the effect of variable injection-pressure and engine-speed at three different throttle positions on both engine Power and NO$_x$ emission in a diesel engine. For this aim, experiments have been conducted for both full and partial loads (throttle positions) on a turbocharger diesel engine with four-cylinder, four-stroke, indirect injection by changing the injection pressure from 100 to 250 bars with interval of 50 bars. The obtained polynomial models based on the experimental data are then used in a Pareto-based optimization approach to find the best possible combination of the engine Power and NO$_x$ emission.

MODELLING USING GMDH NEURAL NETWORKS

By means of GMDH algorithm a model can be represented as set of neurons in which different pairs of them in each layer are connected through a quadratic polynomial and thus produce new neurons in the next layer. Such representation can be used in modelling to map inputs to outputs. The formal definition of the identification problem is to find a function $\hat{f}$ so that can be approximately used instead of actual one, f, in order to predict output $\hat{y}$ for a given input vector $X=(x_1,x_2,x_3,...,x_n)$ as close as possible to its actual output y. Therefore, given M observation of multi-input-single-output data pairs so that

$$y_i = f(x_{i1},x_{i2},x_{i3},...,x_{in}) \quad (i=1,2,...,M)$$

it is now possible to train a GMDH-type neural network to predict the output values $\hat{y}_i$ for any given input vector $X=(x_{i1},x_{i2},x_{i3},...,x_{in})$, that is

$$\hat{y}_i = \hat{f}(x_{i1},x_{i2},x_{i3},...,x_{in}) \qquad (i=1,2,...,M).$$

The problem is now to determine a GMDH-type neural network so that the square of difference between the actual output and the predicted one is minimised, that is

$$\sum_{i=1}^{M} [\hat{f}(x_{i1},x_{i2},x_{i3},...,x_{in}) - y_i]^2 \to \min .$$

General connection between inputs and output variables can be expressed by a complicated discrete form of the Volterra functional series in the form of

$$y = a_0 + \sum_{i=1}^{n} a_i x_i + \sum_{i=1}^{n}\sum_{j=1}^{n} a_{ij} x_i x_j + \sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{k=1}^{n} a_{ijk} x_i x_j x_k + ... \quad (1)$$

which is known as the Kolmogorov-Gabor polynomial (Farlow 1984). This full form of mathematical description can be represented by a system of partial quadratic polynomials consisting of only two variables (neurons) in the form of

$$\hat{y} = G(x_i, x_j) = a_0 + a_1 x_i + a_2 x_j + a_3 x_i x_j + a_4 x_i^2 + a_5 x_j^2 \quad (2)$$

In this way, such partial quadratic description is recursively used in a network of connected neurons to build the general mathematical relation of inputs and output variables given in equation (1). The coefficient a_i in equation (2) are calculated using regression techniques (Nariman-zadeh et al. 2005) so that the difference between actual output, y, and the calculated one, $\hat{y}$, for each pair of x_i, x_j as input variables is minimized. Indeed, it can be seen that a tree of polynomials is constructed using the quadratic form given in equation (2) whose coefficients are obtained in a least-squares sense. In this way, the coefficients of each quadratic function G_i are obtained to optimally fit the output in the whole set of input-output data pair, that is

$$E = \frac{\sum_{i=1}^{M} (y_i - G_i())^2}{M} \to \min \quad . \quad (3)$$

In the basic form of the GMDH algorithm, all the possibilities of two independent variables out of total n

input variables are taken in order to construct the regression polynomial in the form of equation (2) that best fits the dependent observations (y_i, *i=1, 2, ..., M*) in a least-squares sense. Consequently, $\binom{n}{2} = \frac{n(n-1)}{2}$ neurons will be built up in the first hidden layer of the feedforward network from the observations $\{(y_i, x_{ip}, x_{iq})$; *(i=1, 2, ..., M)*$\}$ for different $p, q \in \{1,2,...,n\}$. In other words, it is now possible to construct M data triples $\{(y_i, x_{ip}, x_{iq})$; *(i=1, 2, ..., M)*$\}$ from observation using such $p, q \in \{1,2,...,n\}$ in the form

$$\left[\begin{array}{cc|c} x_{1p} & x_{1q} & y_1 \\ x_{2p} & x_{2q} & y_2 \\ \hdashline x_{Mp} & x_{Mq} & y_M \end{array}\right].$$

Using the quadratic sub-expression in the form of equation (2) for each row of *M* data triples, the following matrix equation can be readily obtained as

$$A\mathbf{a} = Y$$

where $\mathbf{a}$ is the vector of unknown coefficients of the quadratic polynomial in equation (2)

$$\mathbf{a} = \{a_0, a_1, a_2, a_3, a_4, a_5\} \qquad (4)$$

and

$$Y = \{y_1, y_2, y_3, ..., y_M\}^T$$

is the vector of output's value from observation. It can be readily seen that

$$A = \left[\begin{array}{cccccc} 1 & x_{1p} & x_{1q} & x_{1p}x_{1q} & x_{1p}^2 & x_{1q}^2 \\ 1 & x_{2p} & x_{2q} & x_{2p}x_{2q} & x_{2p}^2 & x_{2q}^2 \\ \hdashline 1 & x_{Mp} & x_{Mq} & x_{Mp}x_{Mq} & x_{Mp}^2 & x_{Mq}^2 \end{array}\right]$$

The least-squares technique from multiple-regression analysis leads to the solution of the normal equations in the form of

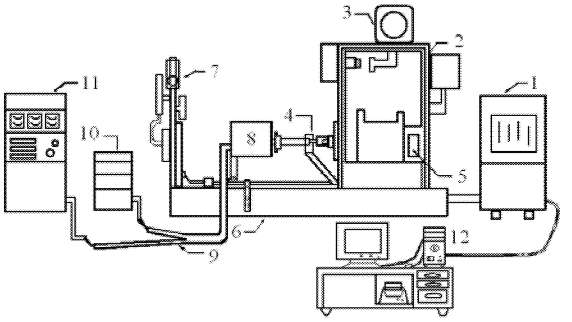
$$\mathbf{a} = (A^T A)^{-1} A^T Y \qquad (5)$$

which determines the vector of the best coefficients of the quadratic equation (2) for the whole set of M data triples. It should be noted that this procedure is repeated for each neuron of the next hidden layer according to the connectivity topology of the network. However, such a solution directly from normal equations is rather susceptible to round off errors and, more importantly, to the singularity of these equations.

Application of GAs and Singular Value Decomposition (SVD) for optimal design of both connectivity configuration and the values of coefficients, respectively, involved in GMDH-type neural networks are given in details in (Nariman-zadeh et al. 2005).

EXPERIMENTAL APPARATUS AND PROCEDURE

Experimental set up and data acquisition system has already been fully described in an investigation by (Çelikten 2003). A brief description is given below. As shown in figure 1 an electrical dynamometer assembled on 4-cylinder and 4-stroke indirect injection diesel engine was employed. In addition, there are two exhaust emission measurement equipment worked separately. One of the equipment was used for CO and NOx measurements (11), the other could be used to measure smoke level (12). After nozzles that are changed pressure adjustment are assembled, they are investigated according to engine performance and emission for different throttle position.



1- control panel	2- device panel	3- air filter
4- shaft	5- dynamometer	6- platform
7- cooling system	8- engine	9- exhaust pipes
10- exhaust emission measurement	11- smoke level measurement	12- computer

Figure 1. Experimental set up.

The computer controlled diesel engine which is connected to the electrical dynamometer was loaded in throttle position of 50%. Engine was tested in range of 4500-1500 rpm with the interval of 500 rpm. In the experiments, torque, Power, brake mean effective pressure (BMEP), specific fuel consumption (SFC), and fuel flow rate were recorded by computer. In addition, emissions and smoke level (or PM) have been measured by exhaust probes connected to the tailpipe. Similarly, these measurements were repeated in throttle positions of 75% and 100%. The experiments performed for the nozzle pressures of 100, 150, 200, and 250 bars. The results of engine performance and emissions measurements were fully presented and studied in (Çelikten 2003). Among all the experimental measurements, engine Power and NOx emission have been employed for investigation in this work.

MODELLING OF POWER AND NO$_X$ USING GMDH

The input-output data pairs used in such modelling involve two different data tables obtained from experiments discussed in previous section. The first table consist of three variables as inputs namely, speed (N), throttle position (TP) and injection pressure (IP) and one output which is Power. The second table consists of the same two variables as inputs and another output which is NO$_x$, of the engine. These tables consist of the total 84 pattern numbers which have been obtained from the experiments to train such GMDH-type neural networks. However, in order to demonstrate the prediction ability of evolved GMDH-type neural networks, the data has been divided into two different sets, namely, training and testing sets. The training set, which consists of 72 out of 84 inputs-output data pairs, is used for training the neural network models using the evolutionary method of this paper. The testing set, which consists of 12 unforeseen inputs-output data samples during the training process, is merely used for testing to show the prediction ability of such evolved GMDH-type neural network models during the training process. In order to genetically design such GMDH-type neural network described in previous section a population of 40 individuals with a crossover probability of 0.85 and mutation probability of 0.07 has been used in 250 generation that no further improvement has been achieved for such population size. The structures of the evolved 2-hidden layer GMDH-type neural network are shown in Figure 2 and Figure 3 for Power and NO_x, respectively. The corresponding polynomial representation of such model for Power is as follows

$$Y1=-71.01+2.33(TP)+0.15IP-0.014(TP)^2-0.00043(IP)^2-0.0003(TP)(IP); \tag{6-a}$$

$$Y2=0.0068+0.0048(N)+0.32(TP)-0.000004(N)^2-0.0076(TP)^2+0.00036(N)(TP); \tag{6-b}$$

$$Y3=-0.00045+0.023(N)-0.068(IP)-0.0000035(N)^2-0.00008*IP\hat{}2+0.000016(N)(IP); \tag{6-c}$$

$$Y4=-0.032+0.013(N)-0.5(Y1)-0.00000541932853*(N)^2-0.018(Y1)^2+0.00085(N)(Y1); \tag{6-d}$$

$$Y5=14.07-0.26(Y2)-0.2(Y3)-0.036(Y2)^2-0.03(Y3)^2+0.1(Y2)(Y3); \tag{6-e}$$

$$Power=-1.3-0.11(Y4)+1.24(Y5)-0.011(Y4)^2-0.037(Y5)^2+0.045(Y4)(Y5); \tag{6-f}$$

Similarly, the corresponding polynomial representation of the model for NO_x is in the form of

$$Y7=-0.0088+0.14(N)-0.41(TP)-0.000056(N)^2-0.018(TP)^2+0.0028(TP)(N), \tag{7-a}$$

$$Y8=-763.49+15.05(TP)+3.4(IP)-0.06(TP)^2-0.005(IP)^2-0.005(TP)(IP), \tag{7-b}$$

$$Y9=-479.72+3.16(IP)+1.7(Y7)-0.005(IP)^2-0.001(Y7)^2-0.00015(IP)(Y7), \tag{7-c}$$

$$Y10=9.55+0.84(Y8)+0.13(IP)+0.0004(Y8)^2+0.0003(IP)^2-0.0007(Y8)(IP); \tag{7-d}$$

$$NO_x=-65.2+0.86(Y9)+0.44(Y10)+0.0035(Y9)^2+0.002(Y10)^2-0.006(Y9)(Y10); \tag{7-e}$$

where *N*, *TP* and *IP* stand for speed, throttle position and injection pressure, respectively.

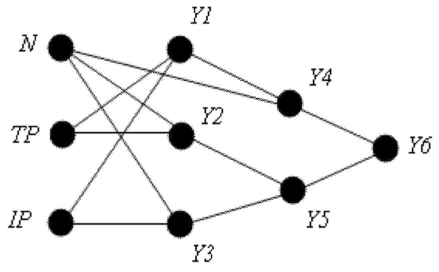


Figure 2. Evolved structure of GMDH-type neural networks for modelling of Power

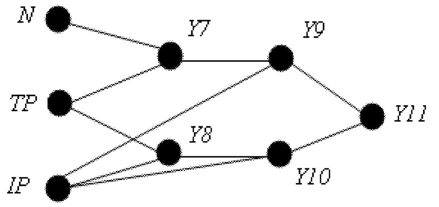


Figure 3. Evolved structure of GMDH-type neural networks for modelling of No_x

The very good behaviour of such GMDH-type neural network models are also depicted in Figure 4 and Figure 5 for testing data of both CO and NO_x, respectively. It is clearly evident that the evolved GMDH-type neural network in terms of simple polynomial equations can successfully model and predict the output of testing data that has not been used during the training process.

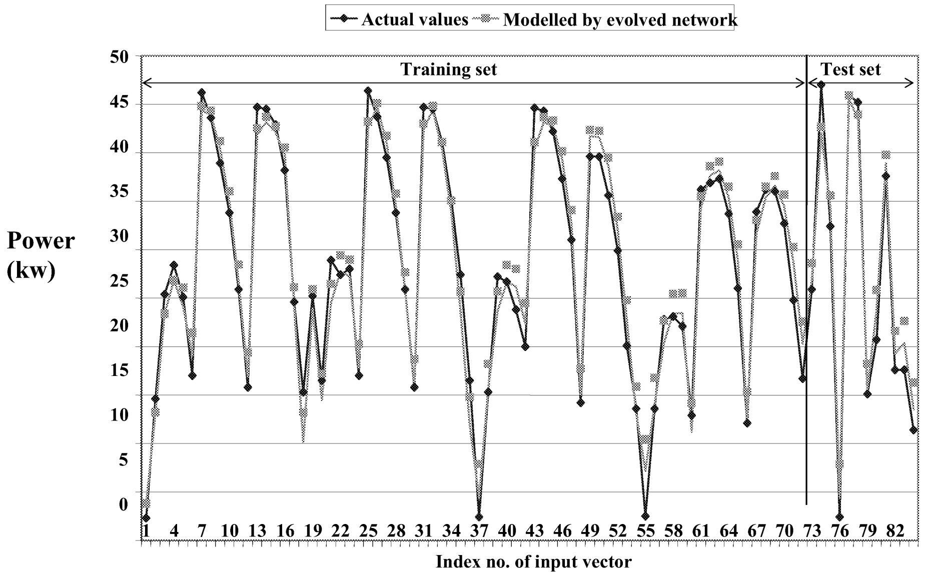


Figure 4. Comparison of experimental data with the evolved GMDH model of Power

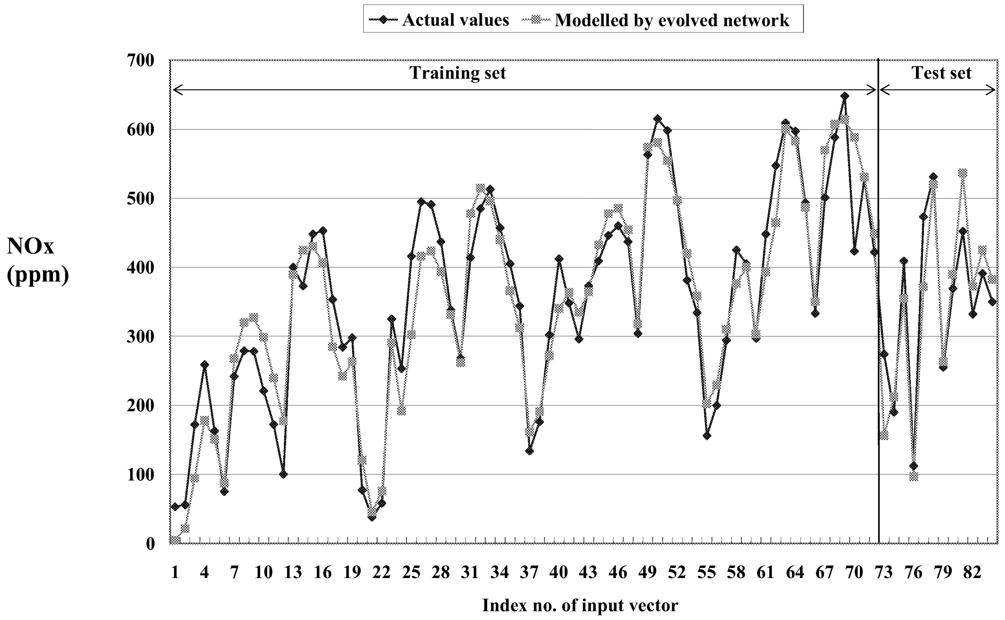


Figure 5. Comparison of experimental data with the evolved GMDH model of No_x

The models obtained in this section can now be utilized for a Pareto multi-objective optimization of engine considering Power and NO_x as conflicting objectives. Such study may unveil some interesting and important optimal design principles that would not have been obtained without the use of a multi-objective optimization approach.

MULTI-OBJECTIVE OPTIMIZATION

Multi-objective optimization, which is also called multi criteria optimization or vector optimization, has been defined as finding a vector of decision variables satisfying constraints to give acceptable values to all objective functions (Atashkari et al. 2005). In general, it can be mathematically defined as:

Find the vector $X^* = \left[x_1^*, x_2^*, ..., x_n^*\right]^T$ to optimize

$$F(X) = \left[f_1(X), f_2(X), ..., f_k(X)\right]^T, \qquad (8)$$

subject to m inequality constraints

$$g_i(X) \le 0 \quad , \quad \mathrm{i = 1 \ to \ m} \ , \qquad (9)$$

and p equality constraints

$$h_j(X) = 0 \quad , \quad \mathrm{j = 1 \ to \ p} \ , \qquad (10)$$

Where, $X^* \in \Re^n$ is the vector of decision or design variables, and $F(X) \in \Re^k$ is the vector of objective functions, which must each be either minimized or maximized. However, without loss of generality, it is assumed that all objective functions are to be minimized. Such multi-objective minimization based on Pareto approach can be conducted using some definitions:

Definition of Pareto dominance

A vector $U = \left[u_1, u_2, ..., u_k\right] \in \Re^k$ is dominant to vector $V = \left[v_1, v_2, ..., v_k\right] \in \Re^k$ (denoted by $U \ \pi \ V$) if and only if $\forall i \in \{1,2,...,k\}$, $u_i \le v_i \ \wedge \ \exists j \in \{1,2,...,k\}$: $u_j < v_j$. In other words, there is at least one which is smaller than whilst the remaining's either smaller or equal to corresponding.

Definition of Pareto optimality

A point $X^* \in \Omega$ (Ω is a feasible region in $\Re^n$ satisfying equations (9) and (10)) is said to be Pareto optimal (minimal) with respect to all $X \in \Omega$ if and only if $F(X^*) \ \pi \ F(X)$. Alternatively, it can be readily restated as $\forall i \in \{1,2,...,k\}$, $\forall X \in \Omega - \{X^*\}$ $f_i(X^*) \le f_i(X) \ \wedge$ $\exists j \in \{1,2,...,k\}$: $f_j(X^*) < f_j(X)$. In other words, the solution X^* is said to be Pareto optimal (minimal) if no other solution can be found to dominate X^* using the definition of Pareto dominance.

Definition of a Pareto set

For a given MOP, a Pareto set $\mathcal{P}^*$ is a set in the decision variable space consisting of all the Pareto optimal vectors $\mathcal{P}^* = \{X \in \Omega | \nexists X' \in \Omega : F(X') \ \pi \ F(X)\}$. In other words, there is no other X' as a vector of decision variables in Ω that dominates any $X \in \mathcal{P}^*$.

Definition of a Pareto front

For a given MOP, the Pareto front $\mathcal{PF}^*$ is a set of vector of objective functions which are obtained using the vectors of decision variables in the Pareto set $\mathcal{P}^*$, that is

$\mathcal{PF}^* = \{F(X) = (f_1(X), f_2(X),....,f_k(X)) : X \in \mathcal{P}^*\}$. In other words, the Pareto front $\mathcal{PF}^*$ is a set of the vectors of objective functions mapped from $\mathcal{P}^*$.

Evolutionary algorithms have been widely used for multi-objective optimization because of their natural properties suited for these types of problems. This is mostly because of their parallel or population-based search approach. Therefore, most of the difficulties and deficiencies within the classical methods in solving multi-objective optimization problems are eliminated. For example, there is no need for either several runs to find the Pareto front or quantification of the importance of each objective using numerical weights. The Pareto-based approach of NSGA-II (Non-dominated Sorting Genetic Algorithm) (Deb et al. 2002) has been used recently in a wide area of engineering MOPs because of its simple yet efficient non-dominance ranking procedure in yielding different level of Pareto frontiers. In this work, modified NSGA-II algorithms are applied for multi-objective optimization of a diesel engine with two objectives, namely, Power and NO_x.

PARETO OPTIMIZATION OF INDIRECT INJECTION DIESEL ENGINE USING POLYNOMIAL MODELS

The conflicting objectives in this study are Power and $NO_{x.}$ These are to be optimized simultaneously with respect to the design variables, namely, speed (N), throttle position (TP) and injection pressure (IP). Evidently, it can be observed that the engine Power is maximized whilst NO_x emission is minimized in the set of objective function ($NO_{x,}$ Power). The evolutionary process of Pareto multi-objective optimization is accomplished by using the NSGA-II approach where a population size of 50 has been chosen in all runs with crossover probability Pc and mutation probability Pm as 0.95 and 0.1, respectively. However, in order to refine the obtained results as non-dominated points, the Pareto front is moved to a better one by using a single-objective optimization technique for each non-dominated points.

The corresponding Pareto front of two objectives Power and NO_x has been shown in Figure 6. It is clear from this figure that choosing appropriate value for speed (N), throttle position (TP) and injection pressure (IP), to obtain a better value of one objective would cause a worse value of another objective. However, if the set of decision variables is selected based on each of the Pareto set, it will lead to the best possible combination of those two objectives. In other words, if any other pair of decision variables, speed (N), throttle position (TP) and injection pressure (IP), is chosen, the corresponding values of the pair of objectives, i.e. Power and NO_x, will locate a point inferior to the Pareto front. Such inferior area in the space of the two objectives shown in Figure 6 is in fact upper/left side. Clearly, there are some important optimal design facts between the two objective functions which have been discovered by the Pareto optimization of the polynomial neural network models obtained using the experimental data. Such important design facts could not have been found without the multi-objective Pareto optimization of those polynomial models. From Figure 6, points C is the point which demonstrate these important optimal design facts. Point C exhibit a very small decrease in the value of Power (about 10%) in comparison with that point of B (best Power) except that its No_x is about 40% better than that of point B. Therefore, point C could be a trade-off optimum choice when considering the minimum values of No_x and maximum values of Power.

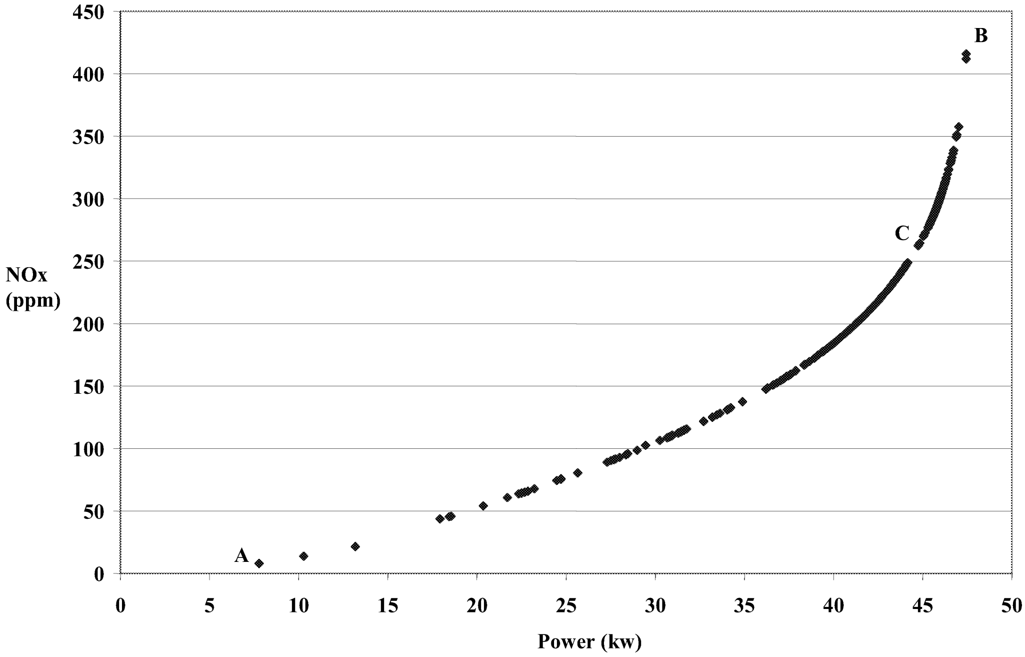


Figure 6. Optimal Pareto front of conflicting objective functions Power and No_x

In Figure 7 the Pareto front obtained from the GMDH-type neural network model (Figure 6) has been superimposed with the corresponding experimental results. It can be clearly seen that such obtained Pareto front lies on the best possible combination of the objective values of experimental data which demonstrates the effectiveness of the approach of this paper both in deriving the model and in obtaining the Pareto front. Besides, the Pareto optimization reveals some interesting and informative design aspects.

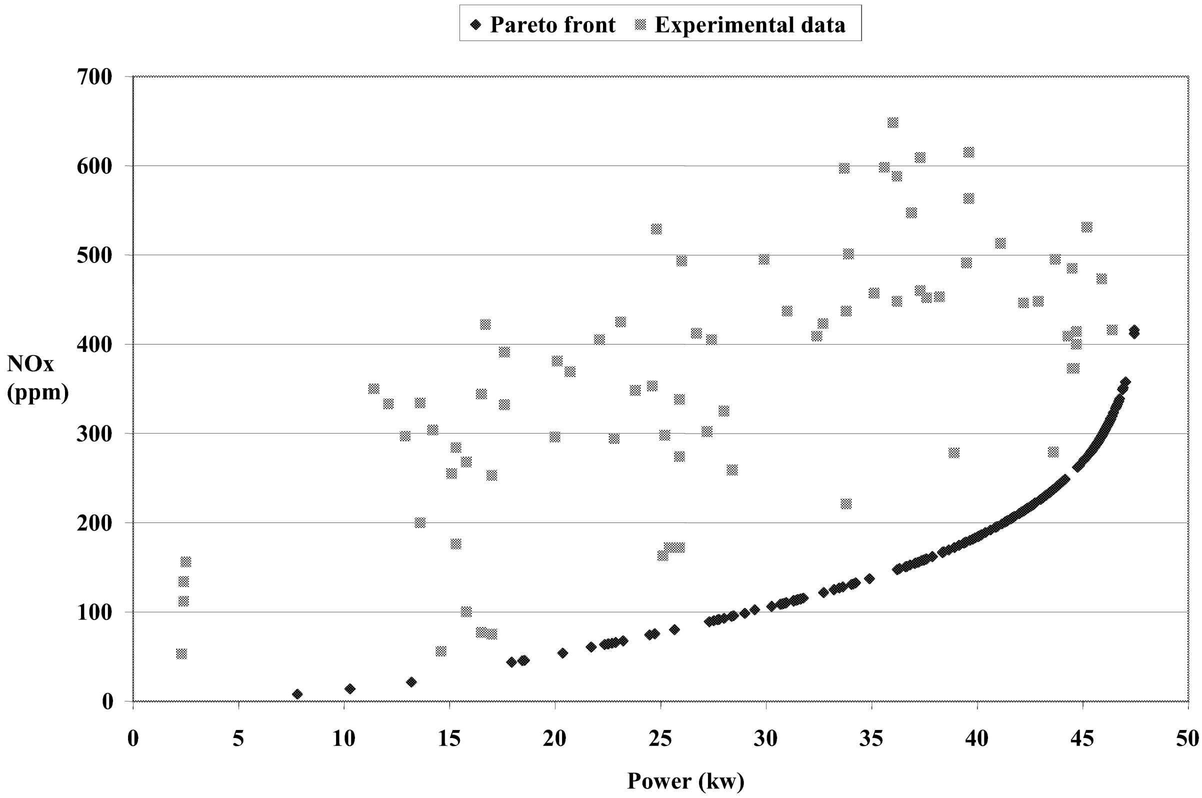


Figure 7. Comparison of Pareto front and experimental data

CONCLUSION

The aim of this paper was to use neural networks for estimation of Power and NO $_x$ emission of an indirect injection diesel engine using different injection pressures, speeds and throttle positions. In this way, two different polynomial relations for the engine Power and NO $_x$ have been obtained by GS-GMDH type neural network employing 72 out of 84 experimental measurements as training data set whilst the remaining 12 were used as testing data set. The overall results showed that the networks were able to successfully model and predict the Power and emissions. Subseqently, the derived polynomial models have then been used in an evolutionary multi-objective Pareto based optimization process so that some interesting and informative optimum design aspects have been revealed for the engine variable with respects to the control variables of engine speed, injection pressure and throttle position. Such combined application of GMDH neural network modeling of experimental input-output data and subsequent non-dominated Pareto optimization process of the obtained models is very promising in discovering useful and interesting design relationships.

REFERENCES

Arora, J.S. 1989. "Introduction to Optimum Design." McGraw-Hill Book Company.

Astrom, K.J. and Eykhoff, P. 1971. "System identification, a survey." Automatica 7, 123-62

Atashkari, K.; N. Nariman-zadeh; A. Jamali; and A. Pilechi. 2005. "Thermodynamic Pareto Optimization of turbojet using multi-objective genethic algorithm." International Journal of Thermal Sciences 44, 1061-1071.

Bruneaux, G. 2001. "Liquid and vapour spray structure in high-pressure common rail diesel injection." Atomization and sprays 11, 533-556.

Celikton, I. 2003. "An experimental investigation of the effect of the injection pressure on engine performance and exhaust emission in indirect injection diesel engines." Applied Thermal Engineering 23, 2051 – 2060.

Deb, K.; S. Agrawal; A. Pratap; and T. Meyarivan. 2002. "A fast and elitist multi-objective genetic algorithm." NSGA-II. IEEE Trans. On Evolutionary Computation 6(2):182-197

Farlow, S.J. 1984. "Self-organizing Method in Modelling: GMDH type algorithm." Marcel Dekker Inc.

Geist, M. 1996. "Reduced exhaust emissions meet the challenge.", Licenses meeting, montreaux, Switzerland, 1-28.

Mueller, J..A. and Lemke, F. 2000. " Self-Organising Data Mining: An Intelligent Approach to Extract Knowledge from Data." Pub. Libri, Hamburg

Nariman-zadeh, N.; K. Atashkari; A. Jamali; A. Pilechi; and X. Yao. 2005. "Inverse modeling of multi-objective thermodynamically optimized turbojet engine using GMDH-type neural networks and evolutionary algorithms." Engineering Optimization, Vol. 37, No. 5, 437-462.

Parlak, A.; H. Yasar; and B. Sahin. 2003. "Performance and exhaust emission characteristics of a lower compression ratio LHR diesel engine." Energy Conversion & Management 44, 163-175.

Parlak, A.; H. Yasar; C. Hasimoglu; and A. Kolip. 2005. “The effect of injection timing on NOx emissions of a low heat rejection indirect diesel injection engine.” Applied Thermal Engineering 25, Issue 17-18, 3042-3052.

Rosenberg, R.S. 1967. “Simulation of genetic populations with biochemical properties.” PhD Thesis, The University of Michigan, Ann Harbor, Michigan.

Srinivas, N. and Deb, K. 1994. “ Multiobjective optimization Using Nondominated Sorting in Genetic Algorithms.” Evolutionary Computation. Vol. 2, No. 3, 221-248.

US-EPA. 1982. “Air quality criteria for oxides of nitrogen.”, Rep. No. EPA-600/8-82-026.

Winterborne, D.E.; D.A. Yates; E. Clough; K.K. Rao; P. Gomes; and J.H. Sun. 1994. “Combustion in high-speed direct-injection diesel-engines a comprehensive study.” Proceedings of the Institution of Mechanical Engineers Part c-Journal of Mechanical Engineering Science 208, 223-240.

Yang, H.C; H.S. Ryou; T.Y. Jeong; and Y.K. Choi. 1996. “Spray characteristics in a direct-injection diesel engine.” Atomization and sprays 6, 95-109.

SUPERVISED FUZZY CONTROL IN THE SIMULATION OF MANUFACTURING SYSTEMS

Karim Tamani*, Reda Boukezzoula*, Georges Habchi**
*LISTIC – Polytech'Savoie **SYMME – Polytech'Savoie
Domaine universitaire
B.P. 806 – 74016 Annecy Cedex
email: {Karim.Tamani, Reda.Boukezzoula, Georges.Habchi}@univ-savoie.fr

KEYWORDS

Simulation, Control Centre, Fuzzy Control, Supervisor.

ABSTRACT

A particular characteristic of a manufacturing system concerns the complexity and the presence of uncertainties through which the difficulties in building analytical models that represent the system from all its major angles. Hence, manufacturing simulation remains one of the most widely used tool to fill this need. The objective of this article is related to the potential improvement of computer simulation as applied to the control of manufacturing system by introducing a two-level fuzzy-logic based control structure. On the lower level of the hierarchy, there is an adaptive fuzzy controller for each specific production module which is synthesized to regulate the flow of the material into a system. On the upper level, a supervisor has the task of coordinating and tuning the local controllers by using performance measurements characterizing the overall system's current behaviour to achieve better performance and restrict the system in admissible domain.

INTRODUCTION

Modern manufacturing systems are characterized by high degree of automation and integration, low levels of work in process inventory, high capital costs, and various forms of supervisory control. While modelling and analysis are important to help ensure good system performance, the integration and complexity of systems often make purely analytic tools difficult to use. Other difficulties come from the uncertainty inherent to data collected for control task and from the necessity of interfacing with human operators.

Hence, simulation is a common way applied to evaluate the performance of a manufacturing system (along with its control system). Traditionally, simulation has been used for offline decision-making. This requires considerable amount of time spent in gathering and analysing data. In order to enhance the capabilities of computer simulation, the task was to find a way of introducing control into simulation by providing generic and applicable concepts (Berchet 2000, Habchi and Berchet 2003). In addition by using practical rules, human experts are the ones that make a manufacturing system works to the desired objective. This leads to the idea to develop a control approach exploiting the behaviour of human experts, that is the emerging field of intelligent manufacturing. The literature offers a wide variety of intelligent techniques for the control of manufacturing systems. In the context of this work, we use the fuzzy theory in the control systems to improve the simulation process. The application of fuzzy control concepts in manufacturing systems has not received much attention until recent years, mainly in the field of scheduling (Angsana and Passino 1994, Custodio et al. 1994, Dadone 1997, Yuniarto and Labib 2005). The problem that we deal is how to regulate the flow of the material into a manufacturing system consisting of a network of resources. The objective is to meet demand for finished products, while guaranteeing stability. The proposed control approach is characterized by two hierarchical levels. In the lower level, there are distributed fuzzy controllers to regulate the production flow in the system, and in the upper level, the supervisor has the task of coordinating and tuning the local controllers, using performance measures characterizing the overall system's current behavior to achieve better performance and restrict the system in admissible domain.

The remainder of this paper is organized as follows. Section 2 introduces the conceptual approach for simulation modelling and control of manufacturing systems, its potentials and limits. In section 3, we present an adaptive fuzzy controller on which our simulation approach is based. Section 4 discusses the supervisory control strategy. In section 5, simulation results are given to illustrate the feasibility of the approach. Concluding remarks are finally given in section 6.

CONCEPTUAL APPROACH FOR MANUFACTURING SYSTEMS

In this section, we present conceptual objects, developed in our laboratory, to model and simulate the resources of a manufacturing system and integrate the control processes in the simulation. Basically, a manufacturing system is divided into three subsystems: physical, informational and decisional. Nevertheless, as simulation models are based on information, we only considered two subsystems: operation and control.

The Production Processing System

In the operation subsystem, we define the Production Processing System (PPS) as a generic object having all structural and functional characteristics of a production resource (Bakalem 1996). It presents the following properties:

- It defines atoms, grouping the natural succeeding of the three fundamental operations of a resource (receiving, processing and supplying);
- It synthesises the resource and its behaviour at the same time;
- It is a recursive structure able to develop models at different levels of abstraction and hierarchy.

The first and second properties describe a PPS standard behaviour consisting in the three-function cycle: *receiving*, *processing* and *supplying*. The third property presents the different states that a PPS could have in a given simulation according to the level of detail needed in the model, and the hierarchical structure allowing the development of models at five different levels (machine, workstation, cell, work-centre, shop).

The Control Centre

In the control subsystem, we define the Control Centre (CC) as an organised and autonomous structure, depending on the company global strategy, having a decisional authority, associated with a controlled entity and having the necessary resources to apply actions and achieve the defined goals within the global framework of the company (Berchet 2000, Habchi and Berchet 2003).
The CC disposes of components: decision-makers, referents, objectives, internal information, external information, performance indicators, measures, actions, control rules, resources. The behaviour of the CC control process is driven by the crisp rules of the form:

IF the control objective given in term of threshold is not satisfied THEN apply the adequat action according to the predefined program based on the cause and effect relation

Figure 1 presents the four main steps in the CC control process:

- *PPS performance evaluation* consists in analysing the measure obtained from the PPS, comparing it with the CC objective, and then concluding if a deviation exists. The main tool used is the performance indicator (e.g., work-in-process inventory measure WIP=12, WIP objective≤10, WIP measure>WIP objective then deviation exists);
- *cause search* concerns the identification of the cause responsible for the PPS deviation. The identification is done by examining lower level performance indicators (e.g., failure rate measure=0.07, failure objective≤0.05, failure rate measure>failure objective, then the responsible cause is failures);
- *action search* consists in the identification of the action able to correct the current deviation of the PPS and prevent future deviations. The seeking of the right action may be done with the help of cause and effect relation (e.g., actions: preventive maintenance, reliability enhancing, quick repairing,...);
- *action applying* concerns the planning and application of actions with the help of the relevant competent resources (e.g., maintenance service).

Advantages and limits

The main advantages of modelling and simulation using the conceptual objects (PPS and CC) can be summarized in the following points:

- The reusability allowed by the generic aspect;
- The possibility of refinement considering the different levels of abstraction;
- The modular modelling property, leading the designer to consider the use of recursive structures of PPS;
- The modelling of operation decision-making within a control process submodel based on feedback loop and performance indicators;
- A clear separation between operation and control submodels.

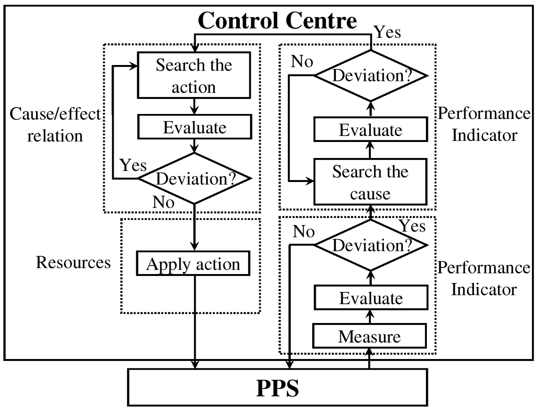


Figure 1: Main steps of the CC control process

In addition, the introduction of the control process into simulation lead to some modifications in the classical simulation process by introducing the following feedback loop – simulation, performance evaluation, action – into the simulation model. The number of simulation runs allowing system optimisation may thus be reduced. However, the current behaviour of the CC is based on crisp rules and there is no learning mechanism taking the past and present values of the performance indicators into account. Furthermore, the human experts are not exploited, and the co-ordination between the different CC are not implemented.

To overcome some of these limits, we introduce a two-level fuzzy-logic based control structure to analyse the CC.

DESIGN OF ADAPTIVE FUZZY CONTROLLER

In this section, we briefly introduce an adaptive fuzzy controller which will be presented in future communication. We consider the in-process inventories, widely known as Work-In-Process (WIP), as a performance measure of manufacturing systems. The control objective is to satisfy the demand and keep WIP as low as possible. This is attempted by constantly regulating the production rate u_i performed at each machine M_i.

A fuzzy controller for each machine M_i is described with the input variables:

- The levels of the upstream and downstream buffers $x_{l,i}, x_{i,k}$ ($l = 1, \ldots, L$ and $k = 1, \ldots, K$) of M_i respectively;
- The production surplus s_i of M_i which is the difference between the cumulative production and demand;
- The state of M_i given by a binary variable α_i (α_i=0 when M_i is down, otherwise α_i=1).

Since the major control objective is to keep the error between the production and demand close to zero, we use an adaptive fuzzy controller based on the Takagi-Sugeno fuzzy model (Boukezzoula et al. 2001). The chosen approach consists in adjusting the conclusion parameter, which provides the fraction of the capacity of the machine devoted to processing.

In the case of a production module composed of a machine M_i, one upstream buffer, and one downstream buffer, the Takagi-Sugeno fuzzy rules describing the controller are:

$R^{(i_1,i_2,i_3)}$: IF $x_{i-1,i}$ is $X_1^{i_1}$ AND $x_{i,i+1}$ is $X_2^{i_2}$ AND s_i is $X_3^{i_3}$ THEN $r_i = \phi(i_1, i_2, i_3)$

where $X_p^{i_p}$ (p=1,2,3) is the i_p^{th} linguistic term associated with the vector of the input variables $x = [x_{i-1,i}\ x_{i,i+1}\ s_i]$, which are the upstream/downstream buffer levels, and the surplus, respectively, while $\phi(i_1, i_2, i_3)$ denotes the real value involved in the rule conclusion. Table 1 shows the fuzzy sets defined for all the input variables. The gains are used to map the actual inputs of the fuzzy system to the normalized universe of discourses (Lee 1990).

Table 1: Linguistic term of the fuzzy sets (E=Empty, A=Almost, N=Normal, F=Full, NEG=Negative, Z=Zero, POS=Positive)

Variables	Fuzzy sets				
$x_{i-1,i}$	E	AE	N	AF	F
$x_{i,i+1}$	E	AE	N	AF	F
s_i	NEG	Z	POS		

The output generated by the fuzzy controller $0 \leq r_i \leq 1$ constantly “decides” how “fast” the machine M_i should produce. In compact form, it is given by:

$$r_i = W \cdot \Phi$$

where $W \in \mathbb{R}^N$ is the row vector composed of the truth degrees of the complete rule base with $N = 5 \times 5 \times 3$ and Φ is the parameter vector of the real values involved in the rule conclusions. The adaptation process involves the adjustment of Φ at each step so that the tracking error (i.e., surplus s) converges to zero. This is applied by using the following algorithms:

$$\Phi(t_{k+1}) = \Phi(t_k) - \eta \cdot W \cdot \mu_i \cdot s_i(t_k)$$

where η is a positive constant value, $\mu_i = 1/\tau_i$ is the maximum rate at which machine M_i can process a part, τ_i the processing time of M_i, and t_k denotes the k^{th} discrete time point.

When the tracking error is satisfied (i.e., surplus close to zero), the controllers keep buffers regulating the machines rates neither full nor empty (Ioannidis et al. 2004). Considering the simple case of one product with one operation, the production rate of machine M_i would be:

$$u_i(t_k) = \frac{r_i}{\tau_i} = r_i \cdot \mu_i$$

As stated in (Angsana and Passino 1994), the choice of the saturation value B (buffer sizes) for every buffer has an influence on the control performance. In the field of fuzzy control, it defines the universes of discourse $[0, B]$ of the buffer level. The optimal buffer sizes are assessed by building safety stocks to compensate future failures. To resolve this problem, we use an iterative approach. The parameter B is initially set to 1. A first simulation is run with this value and the maximum levels on each buffer are used as new values to normalize the B parameters for successive simulations. This procedure is repeated until the B parameters converge.

DESIGN OF SUPERVISORY CONTROLLER

In this section, we describe the supervisor by adapting the approach proposed in (Ioannidis et al. 2004). The objective of the supervisory controller is to restrict the system in the admissible domain of the final surplus; since the surplus is giving a more precise picture of the system's state. If it is negative, customers are not satisfied. If it is positive and has a high value, WIP is high. The supervised control structure is shown in Figure 2.
The input variables of the supervisor are:
The mean surplus of the end product (s), the error variation (ds), and the value of the mean work-in-process (wip). Both the parameters s and ds are used to keep production close to the demand, while the variable wip restricts the number of parts in processing.
The outputs of the supervisor are the correction factors $-1 \leq u_c \leq 1$ and $-1 \leq l_c \leq 1$ of the upper and lower admissible domain (surplus) boundaries, respectively. These correction factors express the percentage by which the domain's limits are altered.
The expert knowledge that describes the supervisory control objective are built on the following assumption; adaptive surplus limits may improve the production performance and guarantee the respect of the specification given in terms of the maximum allowable WIP. It can be summarized in the following statements:

If the WIP is high (low) and the final surplus is positive high (negative high), then reduce (increase) the upper (lower) limit of the admissible domain.

The above knowledge is formally represented as a fuzzy logic rule, as follows:

$R^{(k)}$: IF s is $S^{(k)}$ AND ds is $DS^{(k)}$ AND wip is $W^{(k)}$ THEN u_c is $U^{(k)}$ AND l_c is $L^{(k)}$

The crisp values of the output u_c and l_c, given by defuzzification process, are used to modify the admissible domain bounds according to the following mechanism:

$$s_l = \min\left[s_{l0}(1+l_c), s_u\right], s_u = \max\left[s_{u0}(1+u_c), s_l\right]$$

where s_{l0} and s_{u0} are the lower and upper bounds of the initial domain given in the specification.

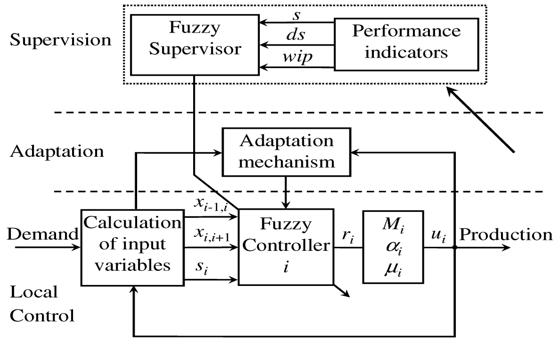


Figure 2: Supervised control structure

SIMULATION TESTING AND RESULTS

Our simulation approach is tested using the example of an assembly line presented in (Ioannidis et al. 2004). The system under consideration consists of five machines producing one product type. The failure and repair rates are equal for all machines. The repair rates are rr_i=0,5 and the failure rates are $p_i = 0,05$. The processing times τ_i $(i = 1, \ldots, 5)$ are chosen as follows:

$$\tau_1 = 2, \tau_2 = 5, \tau_3 = 2, \tau_4 = 1, \tau_5 = 3$$

For comparison purposes, we consider the simple strategy for the CC which decreases the production rate if the inventory falls below some value, increases it if the inventory shoots up above a desired value, and takes no action if it remains within these values. Thus, if inventory i is desired to remain within i_1 and i_2, the crisp rules of the CC will be:

IF $i < i_1$ THEN decrease the production rate of the downstream machine
IF $i > i_2$ THEN increase the production rate of the downstream machine
IF $i_1 \leq i \leq i_2$ THEN no action

All the experiments of the fuzzy control approach have been carried out using MATLAB's FlouLib toolbox (Foulloy et al. 2006), and Simulink, while the conceptual model PPS/CC of the assembly line have been performed with the help of Apollo platform (Habchi and Berchet 2003), in which the described concepts (PPS and CC) have been implemented. The duration of each simulation run is 10000 time units.

Comparative results for the mean work-in-process for various demand patterns are shown in Table 2 and Figure 3. In the case of the conceptual model PPS/CC, all buffer capacities are fixed to 10. In Figure 4, the evolution of the mean WIP in both cases in a simulation run of 10000 time units is presented.
The results show that the fuzzy hierarchical control system achieves a good performance represented by a low WIP. These results are promising; however, further simulation is needed for more complex production models, such as those exhibiting other kinds of uncertainties or disturbances (e.g., demand variations), multi-objectives case, etc., in order to evaluate the robustness of the proposed approach.

CONCLUSIONS

The main purpose of this study was to show the potential improvement of computer simulation as applied to

Table 2: Results for the assembly line test case

Demand rate d (parts/t.u.)	The CC control process	Supervised adaptive fuzzy control
	Mean WIP	Mean WIP
0,05	1,01	0,794
0,08	1,61	0,983
0,1	2,03	1,355
0,15	3,33	2,563
0,18	7,27	4,644

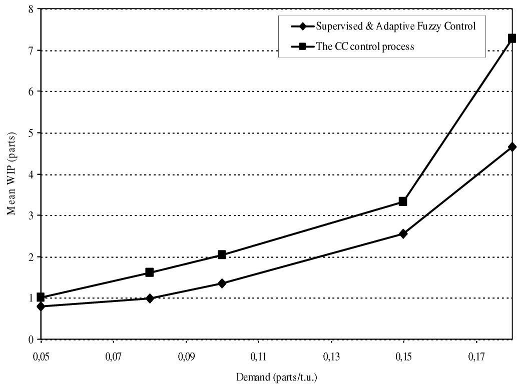


Figure 3: Comparative results of mean WIP with various demands

the control of manufacturing systems through a fuzzy formulation, which bring some intelligence in the computer simulation. Thus, we have introduced a two-level supervised control structure based on the fuzzy theory. The control is distributed, in the sense that each decision is made on the basis of local dynamic information alone. Thus, we have introduced the supervisor that uses actual available data to characterize the overall system's current behavior and then to modify the lower level controllers to ultimately achieve the desired specification. This ensure the coordination between the distributed controllers. For the cases studied, the control algorithm leads to a low WIP compared to the CC control process.

In future work, it would be interesting to consider the case of multi-objectives, including low production lead time, high resource utilization, low tardiness, etc. This leads to multi-criteria aspects of the control. Another interesting extension would be the integration of the proposed approach in the concept of the CC.

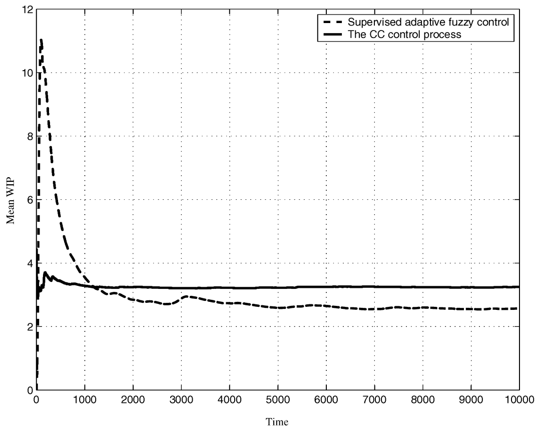


Figure 4: Evolution of mean WIP in the assembly production line with demand d=0,15

REFERENCES

Angsana A. and Passino K.M., 1994. *Distributed Fuzzy Control of Flexible Manufacturing Systems.* In *IEEE Transactions on Control Systems Technology.* vol. 4, 423–435.

Bakalem M., 1996. *Modélisation et simulation orientes objet des systmes manufacturiers.* Ph.D. Thesis, University of Savoie, France.

Berchet C., 2000. *Modélisation pour la simulation d'un systme d'aide au pilotage industriel.* Ph.D. Thesis, University of Grenoble, INPG, France.

Boukezzoula R.; Galichet S.; and Foulloy L., 2001. *Fuzzy Adaptive Control for Nonlinear Systems. Real time implementation for a robot wrist.* In *40th Conference on Decision and Control,Orlando, Florida USA.* IEEE, 4364–4369.

Custodio L.L.M.; Sentieiro J.J.S.; and Bispo C.F.G., 1994. *Production planning and scheduling using a fuzzy decision system.* In *IEEE Transactions on Robotics and Automation.* vol. 10, 160–168.

Dadone P., 1997. *Fuzzy Control of Flexible Manufacturing Systems.* M.S. Thesis, Blacksburg, Virginia.

Foulloy L.; Boukezzoula R.; and Galichet S., 2006. *An Educational Tool for Fuzzy Control.* In *IEEE Transactions on Fuzzy Systems.* vol. 2, 217–221.

Habchi G. and Berchet C., 2003. *A model for manufacturing systems simulation with control dimension.* In *Simulation Modelling Practice and Theory.* vol. 11, 21–44.

Ioannidis S.; Tsourveloudis N.C.; and Valavanis K., 2004. *Fuzzy supervisory control of manufacturing systems.* In *IEEE Transactions on Robotics and Automation.* vol. 3, 379–389.

Lee C.C., 1990. *Fuzzy Logic in Control Systems : Fuzzy Logic Controller – Part I.* In *IEEE Transactions on Systems, Man, and Cybernetics.* vol. 2, 404–418.

Yuniarto M.N. and Labib A.W., 2005. *Optimal control of an unreliable machine using fuzzy-logic control : from design to implementation.* In *International Journal of Production Research.* vol. 21, 4509–4537.

PATH PLANNING AND COGNITIVE MAPS

PATH PLANNING FOR UAVS USING SYMBIOTIC SIMULATION

Farzad Kamrani
Royal Institute of Technology
SE-164 40 Stockholm, Sweden
E-mail: kamrani@kth.se

Marianela Garcia Lozano
Swedish Defence Research Agency
SE-164 90 Stockholm, Sweden
E-mail: garcia@foi.se

Rassul Ayani
Royal Institute of Technology
SE-164 40 Stockholm, Sweden
E-mail: rassul@imit.kth.se

ABSTRACT

The problem of efficient path planning for Unmanned Aerial Vehicles (UAV) with a surveillance mission in a dynamic environment can in some cases be solved using Symbiotic Simulation (S2), i.e. an on-line simulation that interacts in real-time with the UAV and chooses its path. Sequential Monte Carlo Simulation, known also as Particle Filtering (PF) is an instance of such a simulation.

In this paper we describe a methodology and an algorithm to use PF for efficient path planning of a UAV which searches a road network for a target. To verify whether this method is feasible and to supply a tool to compare different methods a simulator is developed. This simulator and its features are presented in this paper as well.

Keywords: Modeling and Simulation, Particle Filtering, Path Planning, Symbiotic Simulation, Unmanned Aerial Vehicle (UAV).

1 INTRODUCTION

Interest for Unmanned Aerial Vehicle (UAV) systems in both civilian and military areas has shown a considerable growth in recent years. Despite diversity in the systems the trend is moving toward increasing autonomy of the UAVs. For example, the United States Air Force Research Laboratory (AFRL) has introduced the notion of Autonomous Control Level, and describes ten such levels, ranging from remotely piloted vehicles to fully autonomous swarms of Unmanned Combat Aerial Vehicles (Clough, 2002). Constructing a framework for highly autonomous UAVs is a demanding task that cannot be tackled using just one science discipline and technique. One natural way to handle this complexity is decomposition of the problem into a hierarchy of manageable sub-problems (Skoglar et al., 2005). For instance, the aerodynamic control of a platform and its long-term path planning belong to different levels of control hierarchy and can be controlled by different subsystems. In this hierarchy we focus on long-term path planning for UAVs. In a static scenario the long-term path planning is primarily deterministic and can be computed off-line (Skoglar et al., 2005). However, in a dynamic situation the long-term path should be re-planned in response to changes in the environment and incoming observation reports. Thus planning in this case, is an on-line process running in parallel with the physical system. Our working hypothesis is that Symbiotic Simulation (S2) (Fujimoto et al., 2002; Lendermann et al., 2005; Low et al., 2005) yields an appropriate decision support tool for efficient utilization of UAVs.

1.1 Problem Statement

The problem, in its most general form, can be formulated as the following. Given an area of responsibility, a set of UAV-borne sensors with different attributes and a prior estimation of the position and capabilities of moving ground targets, the task is to perform a surveillance mission with a minimal total cost. Several parameters like the chosen paths, operational costs of involved UAVs, hostile threats and the value of acquired information determines the total cost of the mission.

Using a state-space approach to model a dynamic system, we use the following notations: $x(t)$ is the state of the observed system in time t, that is a vector holding all information about ground targets in time t. This vector is denoted by x_i when using discrete-time formulation. We use x to refer to the function $x(t)$ itself, i.e. the entire time-evolution of the observed system. Similarly the notations $y, y(t)$ and y_i are used to show the state of the UAV platforms and sensors carried by them. Observations made by the sensors are denoted by $z, z(t)$ and z_i.

With these notations, the problem can be defined as: given the time length of the mission t_k, initial state of the UAVs (sensors) y_0, the initial estimation of the targets $p(x_0)$ and a user-defined objective function $f(y)$, search for $arg \underset{y \in U}{max} \int_0^{t_k} f(y)dt$. The search space is $U = \{y^1 \times \cdots \times y^n\}$, where n is the number of UAVs and y^j is a time dependent vector holding the position and other necessary states of UAV number j and sensors carried by it.

Choice of the objective function f is not a trivial task. However, since the goal of the mission is to track ground targets, we assume that rapidly acquired information has a high value and the importance of an observation decreases by time, i.e. we are looking after such y that gives rise to as much observation as early as possible. Make note that observation z apart from y depends on the unknown state x. Other parameters can be linearly combined with this parameter to construct an objective function that reflects the desire of the UAVs' commander in a specific situation.

Even if path and flight constraints like continuity and differentiability and other constraints on y that reflect the properties of UAVs and sensors, bound the set of all y functions and decrease the size of the set U, this problem is obviously not tractable. However, if we have some rules to choose a limited subset $U' \subset U$, these alternatives can be compared and the best one can be chosen. These rules should be easy to employ and reduce the complexity of the problem to a manageable level according to the computational power available.

UAV path planning is a special case of the sensor resource management which has several research fields and communities as participants, for examples of different approaches to this problem refer to papers in information fusion community (Johansson and Suzić, 2005; Svenson and Mårtenson, 2006; Xiong and Svensson, 2002) and automatic control (Ludington et al., 2005; Skoglar et al., 2005).

2 METHODOLOGY

A system like the one described in 1.1 is partially observable, stochastic, sequential, dynamic, continuous and multi-agent and thus the hardest case to study among different categories of systems (Russell and Norvig, 2003). Since simulation models have the ability to handle uncertainty and can be used for probabilistic reasoning over time, the problem of making strategic decision about UAV motion may with advantage be addressed by using simulation. Due to the constantly changing situation and uncertainty of the acquired information, comparing alternative courses of action in dynamic and complex environments demands a robust control that adapts the model to the changing situation continuously. We suggest that these qualities can be acquired by using Symbiotic Simulations (Fujimoto et al., 2002; Lendermann et al., 2005; Low et al., 2005).

2.1 Symbiotic Simulation

To the best of our knowledge the first time the term "Symbiotic Simulation" was used in a scientific context was in the Modeling and Simulation Seminar at Dagstuhl in 2002 (Fujimoto et al., 2002). At the seminar, Symbiotic Simulation was defined as: "a simulation that interacts with a physical system in a mutually beneficial way". A Symbiotic Simulation is a system which uses an on-line simulation as a means of supporting, controlling or optimizing a physical system. By on-line simulation it is meant a simulation that interacts and exchanges information in real-time with the physical system. This interaction is of benefit for both the physical system and the simulation. The result of the simulation can be used to provide an up-to-date situation awareness or to control the physical system. In addition, the simulation benefits from the continuous supply of the latest data to validate its outputs (Fujimoto et al., 2002). A proposed high level architecture of how simulations and physical system interact is depicted in Figure 1.

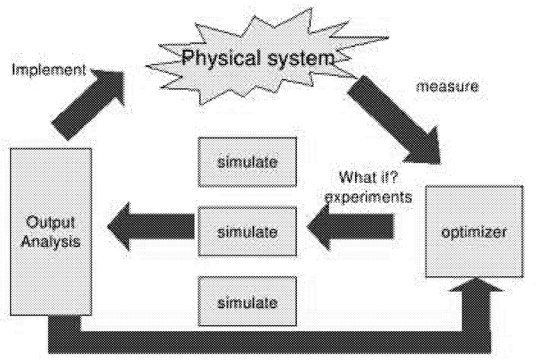


Figure 1: A proposed high level architecture view of how simulations and physical system interact (Fujimoto et al., 2002).

The idea of Symbiotic Simulation obviously has similarities with control systems and the loop depicted in Figure 1 resembles the feedback loop in an automatic controller. However, there are some characteristics that might identify an S2 system. In an S2 system the evaluation of input changes are not performed analytically but by simulations. Since simulations are usually time consuming, interactions between the control and the physical system are more sparse and performed in discrete point of times. Because of the same reason the entire search-space cannot be searched exhaustively, and some "rules" should be used to restrict the search-space to a finite and manageable set of alternative courses of action.

Modeling and simulation is a very vast field and in simulation different methods are deployed depending on the type of the real system and the objective of modeling. The simulation method used here is Sequential Monte Carlo Simulation (Particle Filtering), which is described briefly in Section 2.2. A complete review of Particle Filtering is provided by (Doucet et al., 2001).

2.2 Particle Filtering

In order to analyze a dynamic system using a sequence of noisy measurements, at least two models are required: First, a transition model which describes how the system changes over time and second, a sensor model which relates the noisy measurements to the state (Arulampalam et al., 2002).

Usually these models are available in probabilistic form and since measurements are assumed to be available at discrete times, a discrete-time approach is convenient. In this approach the transition model, $p(x_k \mid x_{k-1})$, gives the conditional probability of the state x_k given x_{k-1}. The sensor model, $p(z_k \mid x_k)$, gives the conditional probability of observation z_k, given the state x_k. We are usually interested in the conditional state of the system, given the sequence of observations $z_{1:k}$, i.e. $p(x_k \mid z_{1:k})$. In general, this probability density function (pdf), may be obtained recursively in two stages, prediction and update. The prediction is calculated before the last observation z_k is available

$$\begin{aligned} &p(x_k \mid z_{1:k-1}) \\ &= \int p(x_k \mid x_{k-1}, z_{1:k-1}) p(x_{k-1} \mid z_{1:k-1}) dx_{k-1} \\ &= \int p(x_k \mid x_{k-1}) p(x_{k-1} \mid z_{1:k-1}) dx_{k-1}. \end{aligned} \quad (1)$$

The first equality follows from
$p(x_k) = \int p(x_k \mid x_{k-1}) p(x_{k-1}) dx_{k-1}$ and the second equality is a result of the fact that the process is Markovian, i.e. given the current state, old observations have no effect on the future state (Arulampalam et al., 2002).

In the update stage the conditional probability $p(x_k \mid z_{1:k})$ is calculated using the prediction result when the latest observation z_k becomes available via Bayes' rule

$$p(x_k \mid z_{1:k}) = \frac{p(z_k \mid x_k) p(x_k \mid z_{1:k-1})}{p(z_k \mid z_{1:k-1})} \quad (2)$$

where the denominator is calculated using

$$p(z_k \mid z_{1:k-1}) = \int p(z_k \mid x_k) p(x_k \mid z_{1:k-1}) dx_k. \quad (3)$$

If the transition model and the sensor model are linear and the process noise has a Gaussian distribution, which is a rather restrictive constraint, these calculations can be performed analytically by using Kalman Filter, otherwise some approximate method such as Particle Filtering (PF) should be used (Arulampalam et al., 2002).

Sequential Monte Carlo Simulation (Particle Filtering) has been shown to be an appropriate tool for estimating the state of a non-linear system with a non-Gaussian process noise, using a sequence of noisy measurements. Intuitively, Particle Filtering is a Monte Carlo Simulation of how the state changes according to the transition model, and to filter the result using the sensor model. Since the system changes over time, this process is repeated in parallel with the real system when new data is received. Even if new observations are not available the prediction stage still can be used to predict the future state of the system. The procedure would be the same with the exception that since future measurements are not known yet, the update stage is not performed.

In particle filter the posterior pdf in each timestep k is represented as a set of n points x_k^i in the state-space and corresponding weights w_k^i, i.e. $p_k^i = \{(x_k^i, w_k^i)\}_{i=1}^n$, where p_k^i is particle number i in $time = k$. The simulation begins with sampling S_0, a set of n particles, from the prior distribution $p(x_0)$, such that

$$S_0 = \{(x_0^i, w_0^i), w_0^i = 1/n\}_{i=1}^n \quad (4)$$

and number of particles in each interval $[a, b]$ is in proportion to $\int_a^b p(x_0) dx_0$.

At each iteration, particles in the set S_{k-1} are updated using the transition model, that is by sampling from

$$p(x_k^i \mid x_{k-1}^i) \quad (5)$$

and when observations arrive the weights are recalculated using

$$w_k^i \propto w_{k-1}^i p(z_k \mid x_k^i). \quad (6)$$

Particles are resampled considering their weights, i.e. they will be sampled with replacement in proportion to their weights and weights are set to $w_k^i = 1/n$.

In a typical surveillance or tracking mission the objective is to estimate the state of the system (position and velocity of the targets) as accurate as possible using a sequence of noisy and uncertain observations received from some kind of sensor carried by a platform (UAV). Particle Filtering and other methods (e.g. Kalman Filter, Extended Kalman Filter) have been extensively used to solve different instances of this problem. In these approaches the transition model describes the physics of motion and the sensor model describes the measurement process.

Here we use the transition model to predict the future state of the system, and use the sensor model to predict the future observations. These simulations are used to search between various suggested UAV paths and choose the one that maximizes an objective function. These decisions are transfered to the control system of the UAV in check points. In these check points new "real" observations are transfered to the particle filter and resampling occurs.

3 S2-SIMULATOR

In order to test and evaluate the idea of using S2 and Particle Filtering for long-term path planning of UAVs in general and testing a special simplified case a simulation tool has been developed. The primary aim of this simulator, called S2-Simulator, is to provide a research tool for experimentation with different methods and techniques and provide indications about scalability and complexity of the algorithms. However in future it may well be developed and function as a prototype with more realistic data. Some system attributes which have been in focus in developing S2-Simulator are simplicity, platform independence and flexibility. S2-Simulator is implemented in Java programming language using object oriented technology.

The S2-Simulator is naturally divided into two parts: First, an emulation of the real world including, terrain, target objects, their plans and UAVs and second, a model of this "real" world that is used in S2-Simulations to improve the behavior of the UAVs.

It should be clear that the information in the real world emulator is not available in the second part, since the real world, e.g. the position and the plan of targets, should not be completely observable for UAV agents. However, for simplicity we can assume that some part of UAVs perception from reality is exact, e.g. the map of the terrain is accurate.

Terrain is modeled as a two-dimensional land and is composed of basic elements like node, road, lake and forest. These elements are user-defined geometric shapes that can easily be added to the terrain using a Graphical User Interface (GUI). The real-world view of the S2-Simulator which is used to generate scenarios is shown in Figure 2.

The probability of existence of targets of different types and the probability that these targets are observed differ in these areas. That is the transition model and the sensor model are dependent of the position in the terrain. Targets can be of different types, such as foot soldier, car, truck and tank. Targets have different characteristics and their behavior varies in different areas of the terrain. A scenario is a configuration of terrain and objects moving on the terrain, during a specified time period. Once a scenario is defined and simulation starts, the operator's interference is not needed or allowed anymore. A scenario should be repeatable, that is, two different executions of a scenario should be exactly the same as long as the seed of the random number generator is not changed.

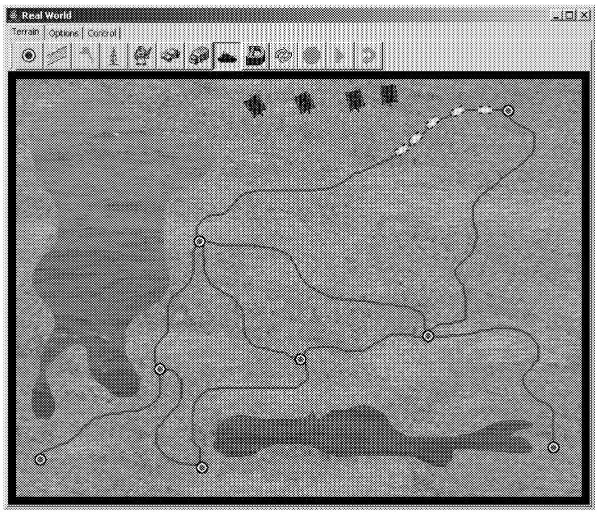
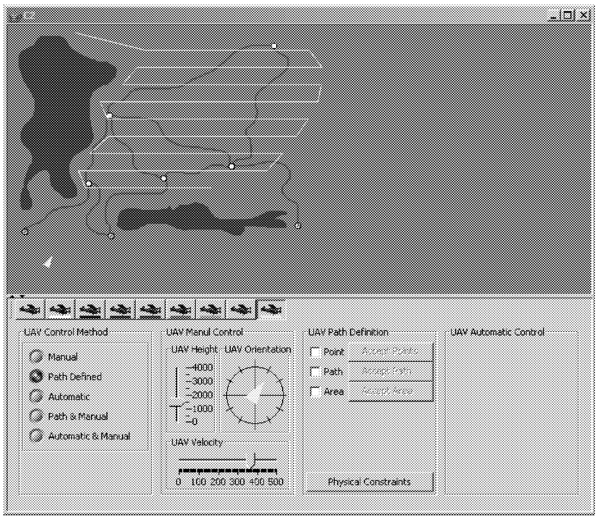


Figure 2: Real-World View of S2-Simulator

UAVs are modeled with reasonable realistic parameters such as velocity, altitude and motion constraints. They are platforms that carry sensors of various kind. The number of available UAVs and the type of sensors are defined prior to the start of the simulation using command and control panel which is shown in Figure 3.


UAV Control Method

○ Manual
● Path Defined
○ Automatic
○ Path & Manual
○ Automatic & Manual

UAV Path Definition

□ Point
□ Path
□ Area

Figure 3: Command and Control of UAVs

3.1 A Test Case

Consider a UAV having the mission of tracking a moving target on a road network where the target, here a vehicle, has a predefined path that is unknown to the UAV. Prior to the tracking mission, some probabilistic information about the initial location of the vehicle is available to the UAV. Furthermore, the movement model of the vehicle as a probabilistic model is specified, i.e. some assumptions about the velocity of the vehicle as well as probabilistic assumptions about the mission of the vehicle.

The road network can be considered as a graph $G = (V, A)$, where V is the set of nodes and A is the set of roads. Each road r^{ij} connecting node n^i to n^j is labeled with a none-negative real number p^{ij} that yields the probability that the vehicle located at node n^i chooses road r^{ij} from the set of outgoing roads from n^i. We denote number of nodes in the road network by N and number of roads starting from each node n^i by M^i. Then $V = \{n^i\}_{i=1}^{N}$ and $A = \{(n^i, n^j, p^{ij})\}_{i=1,j=1}^{N,M^i}$, where the ordered triple (n^i, n^j, p^{ij}) is the road r^{ij} from node n^i to node n^j with the associated p^{ij}. According to probability rules: $\forall n^i \in V, \sum_{j=1}^{M^i} p^{ij} = 1$.

In this simplified example, the number of targets and UAVs are limited to one, and the target strictly follows the road network. The map of the area and road network are known to the UAV and prior to the surveillance the UAV has some information about the probability of existence of the target on a road segment, i.e. the probability $p(x_0)$ is known. Since the motion of the vehicle is bound to the road network, the map of the road network, velocity of the vehicle and the probabilities associated with outgoing roads from a road junction constrain the transition model and specify the movement of the target.

To reduce the complexity of the problem and the search space, we do some assumptions about the movement pattern of the UAV. Since the vehicle is constrained to follow the roads, we assume that the UAV flies approximately above the road network and is inclined to finish surveillance of a road segment (road delimited by two nodes) before it starts flying above a new road. This rule is relaxed if a check point (see 2.2) is in the middle of a road, in this case the rest of the road is considered to be a new road segment and the simulation determines whether the UAV should traverse it first or not.

3.1.1 Transition Model

The transition model which is a probabilistic model of the movement of targets is not exact or complete. It is based on the assumption we have about type, property and goal of targets and information about the environment. The transition model may change during the mission, either as a direct result of new observations or under the influence of external information e.g. analysis of the data in a higher level of data fusion. However, in this simplified example this possibility is ignored and the transition model is a static model that does not change over time. The prior probability $p(x_0)$ is a uniform (or another known) distribution over a road segment. The target has a specific type, with a maximum velocity and the distribution of the velocity is known.

We have a probabilistic assumption about the goal of the target, which in combination with the topology of the road network and quality of the roads can give us the probability that the target follows each outgoing road when it arrives at a crossing. This information is sufficient to specify the transition model. The state-space of the target at time k is $x_k = [r_k, d_k, v_k]$ where r_k is the current road, d_k is the distance the target has moved on road r_k and v_k is the velocity of the target. The noise vector is denoted by $\mu_k = [\xi_k, \rho_k]$, where ξ_k is velocity noise, and $\rho_k \sim U[0, 1]$ is a random number which is used only when the target arrives at a node and reflects the uncertainty of the next chosen road segment by the target (lines 6 to 17 in Figure 4). If the target has not reached the end node of the road, it will stay on the same road segment (lines 4 and 5 in Figure 4). $\Delta t = t_{k+1} - t_k$ is the time elapsed between two successive steps.

```
given: $x_k = [r_k, d_k, v_k], \mu_k = [\xi_k, \rho_k], G = (V, A), \Delta t$
return: $x_{k+1}$
1   propagate($x_k, \mu_k, G, \Delta t$)
2       $d_{k+1} \leftarrow d_k + v_k \Delta t$
3       $v_{k+1} \leftarrow v_k + \xi_k$
4       if $d_{k+1} <$ length($r_k$)
5           $r_{k+1} \leftarrow r_k$
6       else
7           cdf $\leftarrow 0$
8           $n^i \leftarrow$ end node of $r_k$
9           $R \leftarrow$ extract $\{r^{ij}\}_{j=1}^{M^i}$ from $A$
10          while $\rho_k <$ cdf
11              $r^{ij} \leftarrow$ choose $r^{ij}$ with min $p^{ij}$ from $R$
12              $R \leftarrow R \setminus \{r^{ij}\}$
13              cdf $\leftarrow$ cdf $+ p^{ij}$
14          end while
15          $r_{k+1} \leftarrow r^{ij}$
16          $d_{k+1} \leftarrow d_{k+1} -$ length($r_k$)
17      end if
18      return $[r_{k+1}, d_{k+1}, v_{k+1}]$
19  end propagate
```

Figure 4: Transition Model

3.1.2 Sensor Model

The sensor model is generally described by the probabilistic model $p(z_k \mid x_k)$, where x_k is the state of the system, and z_k is the observation. The dimensions of the state z_k are usually, but not necessarily, less than the dimensions of x_k, since the system is not completely observable. We choose to distinguish between the part of the system-state which is not under our control, i.e. the state of the target x_k and

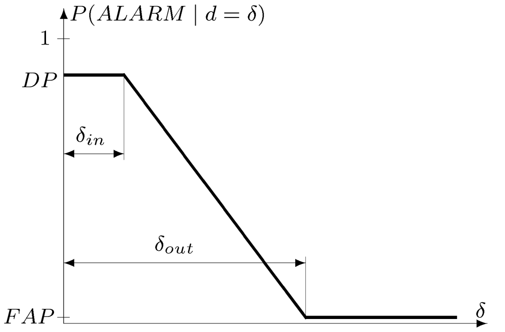


Figure 5: Sensor Model

the state of the sensor y_k. Hence the probabilistic sensor model would be $p(z_k \mid x_k, y_k)$. In our simplified example x_k is specified by the three variables, as described in 3.1.1, which in combination with the terrain map clearly defines the location of the target.

We assume that the video interpretation task is solved by some means, i.e. we have sensors that analyze the incoming video and alarm if they observe any target. Hence the state-space of the observation is binary and $z_k \in [ALARM, \neg ALARM]$. Inspired by (Lichtenauer et al., 2004) we suggest the following model for sensor observations at a standard height.

$$p(ALARM \mid x_k, y_k) = \begin{cases} DP & d \leq \delta_{in} \\ DP - \frac{(DP-FAP)(d-\delta in)}{\delta_{out}-\delta_{in}} & \delta_{in} \leq d \leq \delta_{out} \\ FAP & d \geq \delta_{out} \end{cases} \quad (7)$$

Where DP (detection probability), FAP (false alarm probability), δin and δout are sensor specific and d is the two dimensional distance between the target and the vertical projection of the sensor on the earth, i.e. $d = \sqrt{(x_1^x - x_1^y)^2 + (x_2^x - x_2^y)^2}$. Changing the altitude of the UAV obviously influence the sensor model, but since the sensor model is not the focus of this paper, we assume that UAV has the same altitude.

3.1.3 UAV Path Search

The outline of the algorithm of the main control loop of the UAV is depicted in Figure 6. Given a road network G and a UAV, the objective of this algorithm is to calculate periodically the *best_known_path* and move the UAV repeatedly in time slice Δt along this path. The loop terminates when the the mission time is elapsed and the test at line 6 fails. A feasible chosen time period *time_horizon* divides the mission time to equally long time periods. These time periods determine *check_points* successively which are points that the UAV reaches after a *time_horizon*. A set of particles which always reflects our latest perception of the state of the target is generated at line 4. This set is updated at line 18 when new observations become available and propagated forward at line 19.

```
1   $main\_loop(uav, G, \Delta t)$
2     $s2\_result \leftarrow$ a random path derived from G
3     $check\_point \leftarrow$ start position of uav
4     $\{particle^i\}_{i=1}^N \leftarrow sample(N, p(x_0))$
5     $time \leftarrow 0$
6     while $time < mission\_time\_length$
7       if (uav position = check_point)
8         $best\_known\_path = s2\_result$
9         $uav' \leftarrow$ a model of the uav
10        $\{particle'^i\}_{i=1}^N \leftarrow$ copy of $\{particle^i\}_{i=0}^N$
11        move uav' for time_horizon
12        propagate $\{particle'^i\}_{i=0}^N$ in time_horizon
13        $check\_point \leftarrow$ position of uav'
14        $G \leftarrow G \backslash$ traversed roads by $uav'$
15        $s2\_result \leftarrow s2(G, uav', \{particle'^i\})$
16      end if
17      $z \leftarrow$ read sensors
18      update all $\{particle^i\}_{i=1}^N$ using z
19      propagate all $\{particle^i\}_{i=1}^N$ for $\Delta t$
20      move uav for $\Delta t$ along best_known_path
21      $time \leftarrow time + \Delta t$
22    end while
23  end main_loop
```

Figure 6: Main Loop of the UAV

A model of the UAV including its current state is created (at line 9). This model and a copy of the particles are propagated forward a *time_horizon* at lines 11 and 12. These propagations are our perception of the future. The estimated position of the UAV model after a *time_horizon* yields the next *check_point* at line 13 and the untraversed part of the road network at line 14. These future states are then passed to the S2 simulation, which calculates the best path (line 15) for the UAV after it has passed *check_point*.

The *best_known_path* is calculated by comparing the values of an objective function for a finite set of paths by simulations. These paths are derived from the road network, e.g. by permutating the road segments and letting the UAV survey these roads in sequence (the UAV flies from the end of a road segment to the beginning of the next road segment if these two points are not the same).

The details of the S2 algorithm and the Particle Filter used in the main loop as well as the tests performed will be presented in future papers. However, the preliminary experiments indicate that a UAV supported by the S2 simulation is significantly more efficient in tracking a target compared with a UAV that randomly searches the road network.

4 FUTURE WORK

To complete this work, the first step would be to design tests and verify the results of preliminary experiments.

In the future, more complex scenarios will be studied. The complexity of a scenario may be due to the composition of targets, number of UAVs or more complex terrain models. For instance multi-targets including different types which have a correlated movement and follow a mission are significantly more difficult to model. In the same way, if more than one UAV is involved in the tracking mission a more complicated model would be necessary.

5 CONCLUSION

In this paper, we presented a framework for using Symbiotic Simulation in path planning of UAVs performing a surveillance mission. A special purpose simulator which is developed to test whether this methodology is feasible was described, and a general structure of a simplified test case was outlined. Even though it is necessary to systematically design and perform simulations before any reliable statements could be made, preliminary experiments with the S2-Simulator are promising and indicate that S2 can be used for path planning of a UAV in a tracking mission.

6 ACKNOWLEDGMENTS

This work is part of a project funded by the Swedish Defence Research Agency (FOI). We would like to thank Farshad Moradi, Pontus Svenson and Robert Suzić at FOI for their comments, suggestions and constructive critiques.

References

Arulampalam, S., Maskell, S., Gordon, N., and Clapp, T. (2002). "A Tutorial on Particle Filters for On-line Non-linear/Non-Gaussian Bayesian Tracking". *IEEE Transactions on Signal Processing*, 50(2):174–188.

Clough, B. T. (2002). "Metrics, Schmetrics! How The Heck Do You Determine A UAV's Autonomy Anyway?". In *Proceedings of the 2002 PerMIS Workshop*. NIST Special Publication 990.

Doucet, A., de Freitas, N., and Gordon, N. (2001). *Sequential Monte Carlo Methods in Practice*. Springer Verlag.

Fujimoto, R., Lunceford, D., Page, E., and Uhrmacher, A. M. (2002). "Summary of the Parallel/Distributed Simulation Working Group". In Fujimoto, R., Lunceford, D., Page, E., and Uhrmacher, A. M., editors, *Grand Challenges for Modeling and Simulation, Dagstuhl Report*, pages 49–52.

Johansson, R. and Suzić, R. (2005). "Particle Filter-Based Information Acquisition for Robust Plan Recognition". In *Proceedings of the Eighth International Conference on Information Fusion (FUSION 2005)*, Philadelphia, Pennsylvania.

Lendermann, P., Low, M. Y. H., Gan, B. P., Julka, N., Chan, L. P., Turner, S. J., Cai, W., Wang, X., Lee, L. H., Hung, T., Taylor, S. J., McGinnis, L. F., and Buckley, S. (2005). "An Integrated and Adaptive Decision-Support Framework for High-Tech Manufacturing and Service Networks". In Kuhl, M. E., Steiger, N. M., Armstrong, F. B., and Joines, J. A., editors, *Proceedings of the 2005 Winter Simulation Conference*.

Lichtenauer, J., Reinders, M., and Hendriks, E. (2004). "Influence of the Observation Likelihood Function on Particle Filtering Performance in Tracking Applications". In *Proceedings of the 6th International Conference on Automatic Face and Gesture Recognition*.

Low, M. Y. H., Lye, K. W., Lendermann, P., Turner, S. J., Chim, R. T. W., and Leo, S. H. (2005). "An Agent-Based Approach for Managing Symbiotic Simulation of Semiconductor Assembly and Test Operation". In *AAMAS '05: Proceedings of the fourth international joint conference on Autonomous agents and multiagent systems*, pages 85–92, New York, NY, USA. ACM Press.

Ludington, B. T., Tang, L., and Vachtsevanos, G. J. (2005). "Target Tracking in an Urban Warfare Environment Using Particle Filters". In *Proceedings of IEEE Aerospace Conference*, Big Sky, MT.

Russell, S. and Norvig, P. (2003). *Artificial Intelligence A Modern Approach*. Prentice Hall, second edition.

Skoglar, P., Nygårds, J., Björström, R., Ögren, P., Hamberg, J., Hermansson, P., and Ulvklo, M. (September 2005). "Path and Sensor Planning Framework Applicable to UAV Surveillance with EO/IR Sensors". Technical report, FOI-Swedish Defence Research Agency, SE-581 11 Linköping, Sweden.

Svenson, P. and Mårtenson, C. (2006). "SB-Plan: Simulation-Based Support for Resource Allocation and Mission Planning". In *Proceedings of the Conference on Civil and Military Readiness (CIMI 2006)*, Enköping, Sweden.

Xiong, N. and Svensson, P. (2002). "Multi-Sensor Management for Information Fusion: Issues and Approaches". *Information Fusion*, 3(2):163–186.

ACTION SELECTION IN ROBOTS BASED ON LEARNING FUZZY COGNITIVE MAP AND ANALYSIS OF VARIANCE

Ali Azadeh
Koosha Golmohammadi
Amirhossein Gharehgozli
Department of Industrial Engineering, Department of Engineering Optimization Research and Research Institute of Energy Management and Planning, Faculty of Engineering, University of Tehran, P.O. Box 11365/4563, Iran
E-mail: aazadeh@ut.ac.ir , ali@azadeh.com

KEYWORDS
Fuzzy Cognitive Map, Robots Action Selection, Analysis of Variance

ABSTRACT

One of the main issues in developing automatic response systems especially autonomous robots is selecting the best action among all possible actions. Fuzzy Cognitive Maps (FCMs) aim to mimic the reasoning process of the human. FCMs are able to capture and imitate human behavior by describing, developing and representing models. FCMs are also popular for their simplicity and transparency while being successful in a variety of applications. We developed a novel model that could be used for action selection in robots. This model is constructed on a learning FCM which is relied on improved nonlinear Hebbian Algorithm. We tested our model through a series of practical experiments on the latest version of Soccer Server Simulation3D environment. Our tests involved carefully defined factors to measure the team performance. The significance of the proposed model was verified by analysis of variance and independent t-test.

INTRODUCTION

Action selection in an autonomous robot is the response of the intelligent agent to changes in environment and choosing from a series of possible alternative responses. Action selection is the most important function in the hierarchy of intelligent activities of an autonomous robot. Many researchers in the field of artificial intelligence are investigating the new approaches to improve the action selection process. Improvement in the action selection process in an autonomous agent increases the performance.

Fuzzy Cognitive Maps (FCMs) are well known intelligent analysts (Perusich and McNeese, 2005) because of their simplicity and transparency while being successful in a variety of applications. There are several advantages in using FCM. They are simple and easy to understand so they can be modified when necessary. They are also characterized by flexibility of system design and control, comprehensible structure and operation, adaptability to a given domain, and capability of abstract representation and fuzzy reasoning (Koulouriotis et al, 2003). There are many applications for using FCMs such as: analyzing the extend graph theoretical behavior (Zhang and Chen, 1988), automating human problem solving skills (Juliano, 1990), using as behavior models of virtual world (Dickens and Kosko (1994). modeling and supporting water control systems (Gotoh and Yamaguchi, 1991), designing system model for failure models and effect analysis in process industry (Pelaez and Bowles, 1996), strategy planning and analysis for business behavior of automobile industry (Tsadiras, Margaritis and Mertzios, 1995) diagnosis of diseases (Taber, 1991), analysis of electrical circuits (Styblinski and Meyer, 1991), fault management in distributed network environment (Ndousse and Okuda, 1996) modeling of software development project (Stach and et al, 2004; Stach and Kurgan, 2004) and many others. According to the literature, development methods for FCM are far from being complete and well-defined, mainly because of the deficiencies that are present within the underlying design framework (Koulouriotis, Diakoulakis and Emiris, 2001). We know that the FCM models almost always rely on human knowledge (Aguilar, 2005). Only few researches have investigated and proposed a learning algorithm that is suitable for FCM (Stylios, Groumpos and Georgopoulos, 1999).

In this paper we proposed a structure for action selection based on Fuzzy Cognitive Maps (FCMs). We adapted Nonlinear Hebbian Algorithm (Li and Shen, 2004) to improve the FCM structure. This eliminates the deficiencies of FCM (i.e. dependence on experts' knowledge and opinion, and the potential convergence to undesired steady states) and provides a dynamic behavior and flexibility to the FCM model. The extension of the Hebbian Learning, suggesting nonlinear units has drawn the attention of many researchers in the field (Oja, Ogawa and Wangviwattana, 1991, Oja, 1989). The use of unit with nonlinear activation functions that employ Hebbian rule, may lead to robust principal component analysis (Li and Shen, 2004). We adapted this method to train FCM in our action selection model. To test the capabilities of the new architecture, we used Soccer Server, and collected information on various factors. We compared these factors in many tests through agents that use the new model for their action selection in different situations and simple rule based agents to see the effectiveness of the new method.

The organization of the paper is as follows. Section 2 reviews FCM and learning methods to train FCMs. Section 3 discusses adapting Nonlinear Hebbian Rule for FCM learning in our model. Section 4 introduces the new method for action selection based on learning FCM in agents. Section 5 includes implementation of the new methodology in agents through 3D Soccer Server environment and comprehensive experimental evaluation and discussion of the achieved results. Section 6 concludes the paper and highlights future research discussion.

FUZZY COGNITIVE MAP (FCM)

FCM is a fuzzy model of cognitive map which was introduced by Axelord in 1976 (Axelord, 1976). FCM was firstly introduced by Kosko in 1986 (Kosko, 1986). Kosko stated that "the FCM draws a causal picture and ties facts and things and processes to values and policies and objectives and it lets you predict how complex events interact and play out" (Kosko, 1993). FCMs have two main characteristics: 1) They are fuzzified causal relationships between nodes, and 2) They are dynamic and involve feed back, where the effect of change in one node affects other nodes and in turn can affect the node initiating the change. Some investigators believe that the FCMs are a combination of Neural Networks and Fuzzy Logic (Kosko, 1993). FCMs are systems consisting of concepts and relationships that are adapted from human knowledge (Parsopoulos and et al, 2003). Each concept in FCM has a fuzzy value, and each arc is associated to a fuzzy weight. To obtain the value of a concept, the value of its inputs (concepts) are multiplied by the respective weight; then the results are added and passed in a non-linear fashion. FCMs will clearly demonstrate what happens for a system when some of its components change, or if new concepts are introduced or removed. This is an interesting feature of FCMs for action selection, however the difficulties in formulating and mathematical modeling of FCMs make them costly, time consuming and even impossible (Aguilar, 2005).

Concepts reflect attributes, qualities, characters and sense of the system. Edge strengths are given values on the interval [1, -1]. A value of 1 indicates full causality: an increase in A definitely causes an increase in B, with a value -1 indicating full inverse causality: an increase in A definitely causes a decrease in B. Fuzziness enters the map through fractional values of the edge strengths. Non-integer values indicate partial casualty. These fractional numerical values capture linguistic modifiers such as somewhat, hardly, a little, etc. that might be used in modeling a system and its environment. Dependencies captured by this model can be equivalently expressed by a square matrix, called connection matrix, which stores all weight values for edges between corresponding concepts represented by corresponding rows and columns (Stach and et al, 2005). An example of the FCM along with its connection matrix is shown in section 3 Fig.1. Generally, the value of each concept is calculated by applying the following calculation rule:

$$V_i^{k+1} = f\left(V_i^k + \sum_{\substack{j\neq i \\ j=1}}^{N} V_j^k . W_{ji} \right) \tag{1}$$

Where V_i^{k+1} is the value of concept C_j at time $k+1$, and V_j^k is the value of concept C_j, at time k, W_{ji} is the weight of interconnection between concept C_j and concept of C_i, and f is the sigmoid threshold function. The sigmoid function f belongs to the family of squeezing functions, and usually the following function is used:

$$f(x) = \frac{1}{1+\exp(-\lambda x)} \tag{2}$$

Using the sigmoid function the calculated values of concepts after each simulation step will belong to the interval [0, 1]. Experts determine the structure and the weighted interconnections of the FCM using fuzzy conditional statements (Stylios, Georgopoulos and Groumpos, 1999). More specifically, experts describe the relationships between concepts and they use IF–THEN rules to justify their cause and effect suggestions among concepts. Every expert describes each interconnection with a fuzzy rule; the inference of the rule is a linguistic variable, which describes the relationship between the two concepts and determines the grade of causality between the two concepts. Then the inferred fuzzy weights are aggregated, as they are suggested by experts, and an overall linguistic weight is produced, which is transformed to a numerical weight W_{ji}, and represents the overall suggestion of experts (Papageorgiou, Stylios and Groumpos, 2004). The main deficiencies to manage FCMs are their dependence on human experience and knowledge, and the potential uncontrollable convergence to undesired steady-states. We adapted Nonlinear Hebbian Algorithm for FCM learning as an unsupervised learning algorithm to overcome these deficiencies and improve the efficiency and robustness of the FCM methodology in our model.

THE NONLINEAR HEBBIAN ALGORITHM

Learning Methods of FCM

Learning of FCM involves updating the strengths of causal links (connection matrix). A learning strategy is to improve FCMs by fine-tuning its initial causal link or edge strengths applying training algorithms similar to that of artificial neural networks. There have been proposed some FCM learning algorithms (Papageorgiou, Stylios and Groumpos, 2002, Papageorgiou, Stylios and Groumpos, 2003, Koulouriotis, Diakoulakis and Emiris, 2001, Kosko, 1992). Kosko has initially proposed the Differential Hebbian Learning (DHL), as a form of unsupervised learning, but without any mathematical formulation and implementation in real problems (Kosko, 1986, Kosko, 1992). The Balanced differential learning algorithm for FCM training, based exactly on the DHL, has also investigated (Huerga, 2002). Another proposed approach for FCMs training is the Adaptive Random FCMs based on the theoretical aspects of Random Neural Networks (Aguilar, 2002). This algorithm starts from an initial state and weight matrix of the FCM then adapts the weights to lead the FCM to a desired steady-state. In addition Particle Swarm Optimization (PSO) method has been proposed and used for first time for FCM learning giving very promising results (Papageorgiou and et al, 2004). Furthermore, Evolution Strategies have been used for the computation of the desired Action Concepts' values and system's configuration (Koulouriotis, Diakoulakis and Emiris, 2001).

Two other unsupervised learning algorithms for FCM are: 1) The Nonlinear Hebbian Learning (NHL) which has been investigated to train FCMs (Papageorgiou and et al, 2003). This algorithm is based on the nonlinear Hebbian learning rule and updates only the initially suggested (non-zero) weights of the FCM. These weights are updating synchronously at each iteration step till the termination of the algorithm. The calculated values of weights keep their initial signs and directions, as suggested by experts. All the other weights remain zero and no new interconnections are assigned. 2) Nonlinear Hebbian Learning (NHL) proposed by Sheng-Jun Li and Rui-Min Shen (Li and Shen, 2004). This model starts with initial values of concepts and initial weights (given by expert) however the value of concepts and the zero weights are changed and updated through learning process. The algorithm terminates when the FCM reaches to a desired steady-state. We adapted the recent method to take its great advantages.

Nonlinear Hebbian Algorithm

A weighted learning rule required the definition and calculation of a criterion function (error function) and examine when the criterion function reaches a minimum error that corresponds to a set of weights of neural network (NN). When the error is zero or small enough then a steady state (optimal state) for the NN is reached (Li and Shen, 2004). The steady-state weights define the learning process and the NN model. Thus, the minimization of an objective function is the ultimate goal (Williams and Zipser, 1989, Van der Smagt, 1994, Oja, 1989). The well-known Hebbian learning rule suggests that during the learning session, the neural network receives many different excitements or input patterns as input, and it arbitrarily organizes the patterns into categories. Given random input pattern ***x***, weight vector **w** and output, $y = W^T X$ the criterion function ***J*** maximized by Hebbian's rule may be written as equation (3):

$$J = E\{y^2\} \tag{3}$$

Note that the learning rule here is unsupervised so an additional constraint such as $\|W\| = 1$ is necessary to stabilize the learning rule derived from equation (3). The following optimization problem should be solved adjusting the nonlinear units' weight adaptively:

$$Max: \quad J = E\{Z^2\} \tag{4}$$

$$S.t. \quad \|W\| = 1 \tag{5}$$

To solve equation (4), we employ a stochastic approximate approach, which leads to the following improved nonlinear Hebbian learning rule (Li and Shen, 2004):

$$\Delta W_{ji}^{k} = \alpha_k \Delta W_{ji}^{k-1} + \eta_k z_k \frac{dz_k}{dy_k}\left(x_j^k - W_{ji}^{k-1} y_i^k\right) \tag{6}$$

Note that nonlinear learning rule is seeking a set of weight parameters such that the output of the unit has the largest variance. The nonlinear unit constraints the output to remain within a bounded range, e.g., $z = 1/\left(e^{-y}+1\right)$ (sigmoid function), which limits the output within [0, 1]

Proposed Adapted Learning Algorithm to Train FCM

Output concepts (OCs) stand for the agent's actions in proposed action selection model that interests us, and we want to estimate their values, which represent the final state of the system. We named output concepts action concepts (ACs) in our model. The agent selects action among ACs considering their estimated value. Taking the advantage of the general nonlinear Hebbian-type learning rule for neural networks, the mathematical formalism is introduced incorporating a learning rate parameter and the determination of inputs and ACs. This algorithm relates the values of concepts and values of weights in the FCM models. The NHL rule has the general mathematical form equations (7, 8):

$$\Delta W_{ji}^{k} = \alpha_k \Delta W_{ji}^{k-1} + \eta_k z_k \frac{dz_k}{dy_k}\left(x_j^k - W_{ji}^{k-1} y_i^k\right) \tag{7}$$

$$i, j \neq 1,2,\ldots,s$$

$$Where \qquad z_k = 1/\left(1+e^{-V_j^k}\right) \tag{8}$$

In the proposed method C_1, C_2, ..., C_5 are states in first level. The values of these states are defined through environment parameters, and shouldn't be changed during the learning process. The coefficient α_k, a very small positive scalar factor, is called the impulse parameter; the effect of term $\alpha_k W_{ji}^{k-1}$ is to avoid the process stop at the local optimal place or the plate area of the subject space. α_k has the similar function as the impulse term of back propagation algorithm in artificial neural network. The coefficient η_k is called the learning rate that is also a positive small scalar Factor (Hebb, 1949). We define learning rate parameter η_k to decrease exponentially with simulation cycle, as following Equation (9):

$$\eta_k = b_1.\exp(-\lambda_1.k) \tag{9}$$

Convergence of FCMs depends on the step size η_k decay with time, thus η_k is selected to decrease and the rate of decrease depends on the speed of convergence to the optimum solution and on the weight updating mode. The parameters b_1 and $\lambda_2$are positive learning factors, which are determined using the trial and error method (Huerga, 2002). The learning rate parameter α_k is defined to decrease exponentially with simulation cycle, as following Equation (10):

$$\alpha_k = b_2.\exp(-\lambda_2.k) \tag{10}$$

Where b_2 and λ_2 are positive constants which are determined using trial and error experimental process. These values influence the rate of convergence to the desired region and the termination of the algorithm. High values of parameters η_k and α_k may cause the FCM system to oscillate

(Papageorgiou, Stylios and Groumpos, 2004). The convergence process in desired equilibrium points is very sensitive to the values of η_k and α_k. Thus the suggested bounds for these parameters are within [0, 0.1]. The train weight takes the form:

$$W_{ji}^{k+1} = W_{ji}^{k} + \Delta W_{ji}^{k} \qquad (11)$$

Furthermore, a criterion functions is generated for the proposed algorithm. The criterion function is the minimum of the variation of two subsequent values of (ACs):

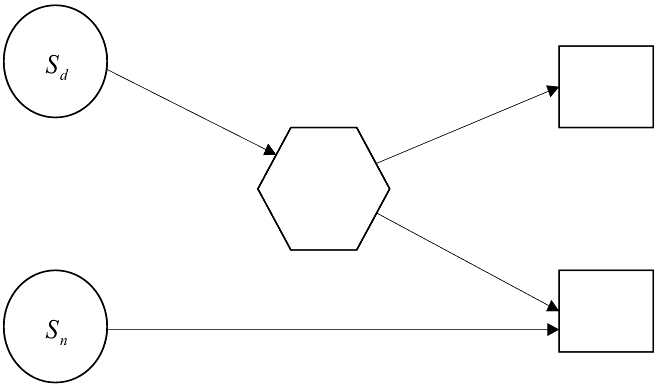
$$\left\lfloor AC_j^{k+1} - AC_j^{k} \right\rfloor < e \qquad (12)$$

This criterion determines when the iteration process of the learning algorithm stops. The term *e* is a tolerance level keeping the variation of the values of ACs as small as possible, here we suggests *e* = 0.001. Through this process and when the process stops, the desired final steady state can be reached, and the final weight matrix W^{Update} is derived.

ACTION SELECTION MODEL

We defined our FCM model in three levels. The first level consists of the primary states. The primary states are basically combinations of the input parameters. The second level consists of states which communicate between the primary states and the third level decision states. The third level consists of decision states. Decision states are a series of possible actions (ACs). The agent is able to autonomously decide and respond to its changing environment. It chooses from actions on a simple basis. Input parameters will affect the overall scores of other states in the second and third level. One state of the third level at a time will achieve the highest score which consequently lead to choose that state as the best action among all possible actions. For each individual action selection an optimal weighted matrix of the three level states is used. This matrix is developed based on an adapted non-linear Hebbian algorithm.

The preliminary weights for matrices and concepts' values are proposed by expert. The agent is capable of updating these original weights based on an adaptive learning algorithm. For each iteration the matrix of weights and values of second and third states are extracted from the updated values of last iteration. Obviously the values of states in first level are specified through the environment. In the following section we illustrate how our model works. We described the flow of procedures in a striker as an agent in Soccer Simulation 3D environment in Figure 1. For the sake of simplicity we defined two input parameters in first level and two possible actions in third level. The striker agent could choose one of actions depending on game situations each time. We defined the weights and concept values as initial values.



○: First level states ⬡: Second level states
□: Third level states

Figure 1: Simple Schema of Proposed Action Selection Model

In our model the agent responds to almost all changes in its environment. Generally in soccer possible situations are divided into two major categories for players. Category "A" includes all situations that the player does not posses the ball. Category "B" is the time that the player is in possession of the ball which we are interested in. The action selection in category "A" is mainly focused on changing the agent's location (position) in the field. However the action selection process in category "B" is mainly focused on what the agent should do with the ball. The value of each state in FCM model is fuzzy. The fuzzy value ranges between 0 and 1 and is mapped with a "mapping function". In our example at any given time the mapping function of the distance of a striker agent in category "B" (in possession of the ball) to the opponent's goal is defined as S_d where d is the actual distance. S_d is defined as equations (13, 14):

$$S_d = 1 - \frac{d}{100} \qquad (13)$$

$$\text{If d> 100 THEN } S_d = 0 \qquad (14)$$

This function maps agent's distance from opponent's goal and S_d is the fuzzified value for its relative state. In the proposed model Number of teammates in pass receiving situation is a parameter which results value of relative concept in first level. Fuzzified value for this concept is defined as S_n. (In our model) *n* is defined according to three major elements at any given time; where 1 is the distance of teammate to the opponent's goal, 2 is the number of opponent agents between the striker agent and the opponent's goal, and finally 3 the distance of teammate to the striker agent. "*n*" is substituted in equations (15, 16) to evaluate S_n:

$$S_n = \frac{n \times 2}{10} \qquad (15)$$

$$\text{If } n > 5 \text{ THEN } S_n = 1 \qquad (16)$$

The values of second and third level states and weight matrix which are updated with adapted nonlinear Hebbian algorithm lead to choose the highest scored state as desired action. However S_d and S_n are specified through environment and don't change in the learning process for each action.

ANALYSIS OF VARIANCE

We implemented the suggested model to our own ARMAN 3D Soccer Simulation RoboCup team which was ranked 5th in RoboCup 2005. We tested our model in the last finalized updated version of the Soccer Simulator 3D environment. This version was used in RoboCup 2005 and is going to be used in RoboCup 2006 in Bremen, Germany. We used ARIA 3D Soccer simulation RoboCup team which was ranked 1st in RoboCup 2005 for our tests. In the primary test setting ARIA faced the modified ARMAN team (model implemented version) for 10 complete matches. Then, ARIA faced the original ARMAN team for another 10 consecutive complete matches. The overall results were encouraging for implementing the new model.

The ultimate criterion of a good team performance is wining the game. We also collected detailed information regarding a group of incidents in the games. Those incidents in our judgment were correlated of dominancy of a team and included: goal advantage, shots on target, number of corners and overall rate of ball possession. Figure 2 shows the graphical illustration of goal differences.

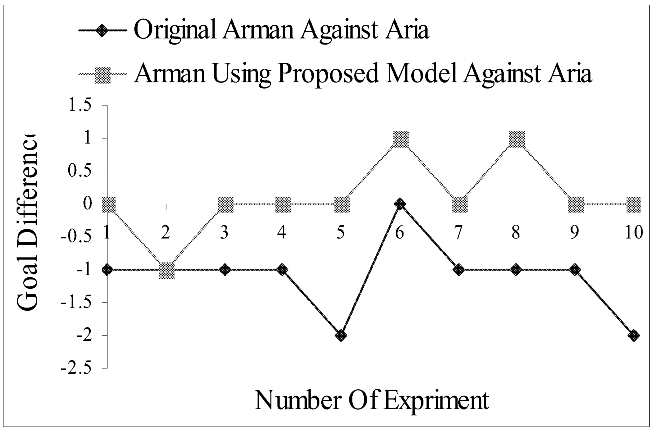


Figure 2: Comparison of Goal Advantage

The results of experiments from the original Arman against Aria was tested (μ_1) with Arman using the proposed model against Aria (μ_2) via analysis of variance (ANOVA) with respect to goal advantage, rate of ball possession, number of shots on target and number of corners. The experiment was designed and tested independently. There was no need to control the variability arising from extraneous sources because of independency assumption (Montgomery and et al, 1999). The results are shown in Table 1. The test of hypothesis is:

$$H_0: \mu_1 = \mu_2 \qquad (17)$$
$$H_1: \mu_1 < \mu_2$$

Where μ_1 and μ_2 are the average estimates obtained from the two models, respectively. It can be seen from Table 1 that at $\alpha = 0.01$ the null hypothesis is rejected for the above four cases because the tabulated $f_{\alpha,18}$ values are relatively larger than the estimated f values.

Table 1: The Results of Analysis of Variance

Treatments	Levene's test for equality of variance		t-test for equality of means			
	F	P-value	t	P-value	95 % confidence interval	
					Lower	Upper
Goal advantage	0.00	1.00	-4.73	1.00	-1.73	-0.67
Rate of ball position	0.32	0.58	-10.43	1.00	-18.74	-12.46
Number of shots on goal	0.93	0.35	-3.88	0.99	-2.01	-0.59
Number of corners	1.70	0.21	-2.61	0.98	-1.28	-0.13

CONCLUSION

In this paper we adapted nonlinear Hebbian algorithm to take its advantages in our action selection model. The proposed model for action selection based on FCM gives the capability of selecting the best action among possible actions to autonomous robots. We implemented the suggested model to our own ARMAN 3D Soccer Simulation RoboCup team which was ranked 5th in RoboCup 2005. We tested both modified Arman and original Arman against Aria which ranked 1st in RoboCup 2005. The overall results were very promising and encouraging. The future work will concern extending the proposed action selection model to other complex systems and autonomous robots. Investigating new approaches for learning methods to train FCMs will improve action selection methods greatly.

REFERENCES

Aguilar, J. 2005. "A Survey about Fuzzy Cognitive Maps." International Journal of Computational Cognition, Vol. 3, no. 2, pp. 27-33.

Aguilar, J. 2002. "Adaptive Random Fuzzy Cognitive Maps." In F.J. Garijio, J.C. Riquelme, M. Toro (Eds.), IBERAMIA, Edmonton, Alberta, Canada, Lecture Notes in Artificial Intelligence, 2527, Springer-Verlag, Berlin, Heidelberg, pp. 402–410.

Axelord, R. 1976. Structure of Decision, the Cognitive Maps of Political Elite. Princeton University Press, Princeton, NJ.

Dickens, J.A. and B. Kosko. 1994. Fuzzy Virtual World. AI Expert 25-31.

Gotoh, K. and T. Yamaguchi. 1991. "Fuzzy Associative Memory Application for Plant Modeling." Proceeding of 1991 International Conference on Artificial Neural Networks, Espoo, Finland, pp 1245-1249.

Hebb, D.O. 1949. The Organization of Behavior: A Neuropsychological Theory, John Wiley, New York.

Huerga, A.V. 2002. "A Balanced Differential Learning Algorithm in Fuzzy Cognitive Maps." In Proceedings of the Sixteenth International Workshop on Qualitative Reasoning, Barcelona, Spain.

Kosko, B. 1993. Fuzzy Thinking: the New Science of Fuzzy logic. New York, Hyperion.

Kosko, B. 1992. Neural Networks and Fuzzy Systems, Prentice-Hall, Englewood Cliffs.

Kosko B. 1986. "Fuzzy Cognitive Maps." International Journal of Man-Machine Studies 24, pp. 65-75.

Koulouriotis, D.E.; I.E. Diakoulakis; and D.M. Emiris. 2001. "Learning Fuzzy Cognitive Maps Using Evolution Strategies: A Novel Schema for Modeling and Simulating High-Level Behavior." IEEE Congress on Evolutionary Computation (CEC2001), pp. 364-371.

Koulouriotis, D.E.; I.E. Diakoulakis; D.M. Emiris; E.N. Antonidakis; and I.A. Kaliakatsos. 2003. "Efficiently Modeling and Controlling Complex Dynamic Systems Using Evolutionary Fuzzy Cognitive Maps (invited paper)." International Journal of Computational Cognition, vol. 1, no. 2, pp. 41-65.

Juliano, B.J. 1990. "Fuzzy Cognitive Structures for Automating Human Problem Solving Skills Diagnoses." Proceeding of the 9' Annual NAFIPS Conference, pp311-314.

Li, S. J. and R. M. Shen. 2004. "Fuzzy Cognitive Map Learning Based On Improved Nonlinear Hebbian Rule." In proceedings of the Third International Conference on Machine Learning and Cybernetics, Shanghai.

Lin, C.T. and G. Lee. 1996. Neural Fuzzy Systems: A Neuro-Fuzzy Synergism to Intelligent Systems, Prentice-Hall, Upper Saddle River.

Montgomery, D. 1999. Design & Analysis of Experiments, John Wiley & Sons, New York.

Ndousse, T.D. and T. Okuda. 1996. "Computational Intelligence for Distributed Fault Management in Networks Using Fuzzy Cognitive Maps." Proceedings of the IEEE International Conference on Communications Converging Technologies for Tomorrow's Application, pp. 1558-1562.

Oja, E.; H. Ogawa; and J. Wangviwattana. 1991. Learning in Nonlinear Constrained Hebbian Networks. In T. Kohonen et al. (Eds.), Artificial Neural Networks. Amsterdam, North-Holland, 385-390

Oja, E. 1989. "Neural Networks: Principal Components and Subspaces." International Journal of Neural Systems, Vol. 1, 61-68.

Papageorgiou, E.I.; C.D. Stylios; and P. P. Groumpos. 2002. "Activation Hebbian Learning Rule for Fuzzy Cognitive Maps." In Proceedings of 15th IFAC World Congress of International Federation of Automatic Control, Barcelona, Spain.

Papageorgiou, E.I.; C.D. Stylios; and P.P. Groumpos. 2003. "Fuzzy Cognitive Map Learning Based on Nonlinear Hebbian Rule." In T.D. Gedeon, L.C.C. Fung (Eds.), 16th International Conference on Artificial Intelligence, AI, Lecture Notes in Artificial Intelligence, 2903, Springer-Verlag, Berlin Heidelberg, pp. 254–266.

Papageorgiou, E.I.; K.E. Parsopoulos; P.P. Groumpos; and M.N. Vrahatis. 2004. "Fuzzy Cognitive Maps Learning Through Swarm Intelligence." In International Conference Artificial Intelligence and Soft Computing, (ICAISC 2004), Zakopane, Poland, Lecture Notes in Computer Science.

Papageorgiou, E.I.; C.D. Stylios; and P.P. Groumpos. 2004 "Active Hebbian Learning Algorithm to Train Fuzzy Cognitive Maps." International Journal of Approximate Reasoning 37, 219–249.

Parsopoulos, K.E.; E.I. Papageorgiou; P.P. Groumpos; and M. N. Vrahatis. 2003. "A First Study of Fuzzy Cognitive Maps Learning Using Particle Swarm Optimization." Proceedings of the IEEE 2003 Congress on Evolutionary Computation, pp. 1440-1447.

Pelaez, C.E. and J. B. Bowles. 1996. Using Fuzzy Cognitive Maps as A System Model for Failure Models and Effect Analysis. Information Science 88, 177-179.

Perusich, K. and M. D. McNeese. 2005. "Using Fuzzy Cognitive Maps as an Intelligent Analyst." IEEE International Conference on Computational Intelligence for Homeland Security and Personal Safety Orlando, FL, USA.

Stach, W.; L. Kurgan; W. Pedrycz; and M. Reformat. 2005. "Evolutionary Development of Fuzzy Cognitive Maps." The IEEE International Conference on Fuzzy Systems, 619-624.

Stach, W.; L. Kurgan; W. Pedrycz; and M. Reformat. 2004. "Parallel Fuzzy Cognitive Maps as a Tool for Modeling Software Development Project." Proceedings of the 2004 North American Fuzzy Information Processing Society Conference (NAFIPS'04), pp. 28-33, Banff, AB.

Stach, W. and L. Kurgan. 2004. "Modeling Software Development Project Using Fuzzy Cognitive Maps." Proceedings of the 4th ASERC Workshop on Quantitative and Soft Software Engineering (QSSE'04), pp. 55-60.

Stylios, C.D.; P.P. Groumpos; and V.C. Georgopoulos. 1999. "A Fuzzy Cognitive Maps Approach to Process Control of Systems." Journal of Advanced Computational Intelligence, Vo1.3, No.5, pp. 409-417.

Stylios, C.D.; V. Georgopoulos; and P.P. Groumpos. 1999. "Fuzzy Cognitive Map Approach to Process Control Systems." Journal of Advanced Computational Intelligence 3 (5) 409–417.

Styblinski, M.A. and B.D. Meyer. 1991. "Signal Flow Graphs Versus Fuzzy Cognitive Maps in Application to Qualitative Circuit Analysis." International Journal of Man-Machine Studies, vol. 35, pp.175-186.

Taber, R. 1991. "Knowledge Processing with Fuzzy Cognitive Maps." Expert Systems with Applications, Vol. 2, pp. 83-87.

Tsadiras, A.; K. Margaritis; and B. Mertzios. 1995. Strategy Planning Using Extended Fuzzy Cognitive Maps. Studies in Informatics and Control 4(3) 237-245.

Van der Smagt, P.P. 1994. Minimization Methods for Training Feed Forward Neural Networks. Neural Networks 7 (1), 1–11.

Williams, R.J. and D. Zipser. 1989. A Learning Algorithm for Continually Running Fully Recurrent Neural Networks. Neural Computation 1 (2), 270–280.

Zhang, W. and S.S. Chen. 1988. "A Logical Architecture for Cognitive Maps." In Proceedings 2nd IEEE International Conference on Neural Networks, Vol2, San Diego, A 24-27, pp381-388.

BIOLOGICAL SIMULATION

GEOMETRIC HIERARCHICAL DATA ORGANISATION IN THE MODELLING OF THE CEREBELLUM

Omar Bennani[1], P. Chauvet[2], F. Jouen[1], G.A. Chauvet[1]

[1] Laboratoire Développement et complexité, Ecole Pratique des Hautes Etudes
41 rue G. Lussac, F75005 Paris (France)

[2] Institut de Mathématiques Appliquées, UCO, Angers (France).

KEYWORDS
Cerebellum, hierarchical modelisation, neural density, synaptic density-connectivity.

ABSTRACT

The modelling and simulation of a realistic nervous tissue is difficult because of the number of implied cell types (neuronal and glial), the topology of the networks, the various and heterogeneous molecular mechanisms. The MTIP (Mathematical Theory of Integrative Physiology) is used for a new modelling approach based on a representation in terms of functional interactions and a formalism (S-Propagator) related to *n*-level field theory. Cell densities and synaptic density-connectivity are the two geometrical parameters that describe the anatomy of the system. This work presents the passage from the theoretical description of the biological system to the geometrical description of these anatomical parameters, and then to its computing implementation in the general case. The specific case of the cerebellum, with a simplified tissue reduced to neuron types, is presented for which we have chosen cylindrical coordinates reference system.

I. INTRODUCTION

The major problem in physiology is related to the lack of formalisation for physiological mechanisms and the integration of the huge amount of data. This is specifically true for the sensorimotor system (Thompson and Krupa 1994) and development studies (Jouen and al. 2002). The G.A. Chauvet MTIP (Chauvet 1993; Chauvet 1999; Chauvet 2002) has been developed to offer a new approach in physiological system modelling. In this approach the specific conceptualization of a given biological system leads to a general mathematical formulation, because of a unifying theoretical framework that provides a common formalism for different phenomena at different levels of a hierarchical system. Specifically, applied to the real neural network (Chauvet and Chauvet 2002), it gives an interpretation of biological processes with a highly integrated view, using techniques and tools of continuous mathematical analysis. It consequently has both practical and theoretical significance in biological research. The formulation then results in the simulation of new emergent properties, such as those produced at the higher levels of the organism by changes occurring at the molecular level.

P. Chauvet developed in collaboration with J.M. Dupont the research simulation tool based on the MTIP, by using object oriented programming which is, and integrates models defined themselves The computing system couples elementary models that correspond to elementary physiological mechanisms. The result is the emergence of the physiological function, e.g. learning. Thus, an integrated model is made of coupled sub-models (generally dynamical systems), each sub-model being defined in a given time scale and on a given region of a hierarchical metric space where are located these physical structures.

In the framework of the MTIP, the structure consists in the anatomy of the system, i.e. an organized ensemble of physical structures. Geometrically, the structures consist in two parameters, the density of neurons and the density-connectivity of synapses defined in the physical space, and which may be state variables in the case of development. In this paper, we present how to pass from the theoretical significance of these anatomic parameters to their geometrical description, and then to their computing implementation. Because the application provides a comprehensive treatment of the integration of cellular and molecular mechanisms of learning and memory in the sensorimotor system, the specific case of the cerebellar cortex will then be presented as an illustration of the method. Although our work is dealing with nervous network, it should be noted that it does not affect the general implications of the approach in the whole biological field.

II. THEORETICAL BASES

The MTIP describe couplings between biological structures using (i) a representation in terms of hierarchical functional interactions and (ii) a specific mathematical formalism, the S-Propagator, to traverse the levels of organization. In this representation, a biological system may be viewed as a hierarchical mathematical graph of functional interactions between biological structures.

1) Functional interaction

The functional interaction acts from one structural unit, the source, onto another, the sink, by means of a signal, causing the sink to produce a new signal at a distance.

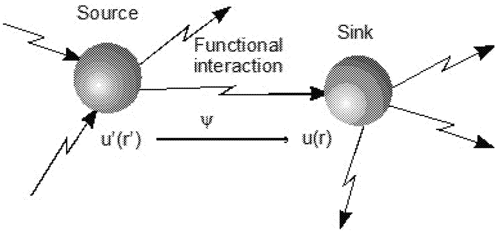


Figure 1: The functional interaction ψ is emitted by the neuron-source u' at $r'$and acts into the neuron-sink u at r (Chauvet 2002)

Because structures are hierarchically organized, we may consider *spaces of units* as abstract spaces included each in the other (Russian dolls). For example, in the nervous system, the action of one unit on another, which is the action of one neuron at r' (a volume) on another at r is represented by ψ emitted by the axon hillock at r' (a point) to the other at r. Let us call u' the source at r' in this abstract mathematical space, the space of units U derived from the physical space, u the sink in the same space at r, and ψ the interaction from u' to u. The functional interaction will be represented as (see Figure 1): ψ.

2) Functional interaction properties

The *non-symmetry* of biological functional interaction represents the action from one structural unit on another situated at a distance, i.e. from a source to a sink, although not directly from source to sink. This interaction is unidirectional since the molecule or signal, emitted at a given level of organization, will have no *direct* retroaction from sink to source, because of the transformation in the sink.

The second important property is the *non-locality* of the functional biological interaction, i.e. a space property according to which the system depends on mechanisms that are located elsewhere in the space because the product emitted is transported at finite speed from source to sink**.** The two constraints of continuous representation of state variables and hierarchy of the system result in non-locality. One difficulty in the hierarchical approach arises from the fact that a local phenomenon in time and space at a given level is not local at lower levels. Thus, local variables for a given unit, e.g. at r in *Figure 1*, depend not only on other local variables but also on global variables for the other units connected to it, through the sub-units included in units

This paper addresses the geometrical part of this problem. More specifically, we show the passage from the theoretical description of the biological system in mathematical abstract spaces to the geometrical description of anatomical parameters, and then to its computing implementation in the general case.

III. GEOMETRICAL PARAMETERS THE MTIP

1) Hierarchical spaces

The physical extension of the structural units in the physical space may be projected onto the *x*-axis, then onto the *r*-space where they are represented by a point (Fig. 2). Inside units "neurons" exist other structures at the lower level, which are the "synapses" *as concerns the function of transfer between the two neurons*. The neural tissue is thus considered as an ensemble of axon hillocks and a neuron is abstracted as an ensemble of synapses. As shown previously, neurons are condensed into the structural units $u(r)$ at r in the *r*-space, and synapses are condensed into the structural units $u(s)$ at s in the *s*-space. Thus *spaces of units are abstract spaces in which defined physical structures capable of action may be considered as points.*

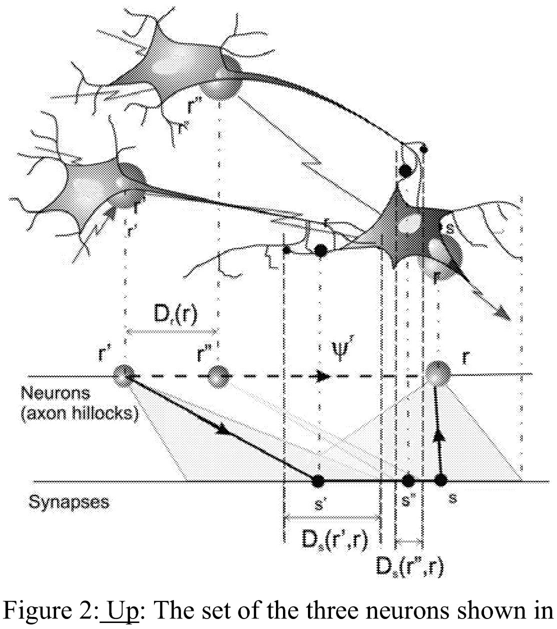


Figure 2: Up: The set of the three neurons shown in terms of functional interactions. Bottom: the functional interaction goes from the neuron at r' in the ensemble $D_r(r)$ of the neurons connected to r at the higher level of neurons, onto the neuron at r, through the synapses at the lower level. For instance, s' is a synapse that makes the communication between the neuron at r' and the current neuron at r, i.e. an element of the ensemble

$D_s(r',r)$. They are structural discontinuities for the neurons. The local effect at r results from effects at r', r'',…, i.e. effects located at a distance through s', s'', … This is non-locality (Chauvet 2002).

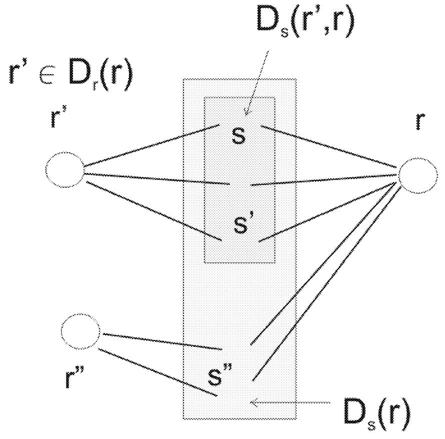
2) Density of structural units and density-connectivity of units between two levels of organization

The MTIP considers a biological system as a combination of functional interactions that act dynamically in a continuous time and space. Thus, Sources and sinks are locally defined by their density, a function ρ, and the connectivity π between them. In the case of neuronal tissues neurons will be represented in the abstract space of neurons D_r by density $\rho(r)$ and the synaptic connectivity between neurons in r and r' by the function $\pi(s;r,r')$ in the abstract space of synapses D_s .(see Fig.3).

The density of structural units $\rho(r)$ is easy to conceive, which is not the case for the density-connectivity because of the three variables s, r and r' that vary in the different abstract space D_r and D_s.

From the physiological point of view, the action potential propagates from the source to the sink, i.e. must pass through the axon in r', the synapse in *s*'(*r*), the cell body from *s*'(*r*) to the axon hillock at r, because the axon hillock is chosen as the source (and thus the sink). The functional interaction goes from the neuron at $r(x',y',z')$ in the ensemble $D_{r(x,y,z)}$ of the neurons connected to r at the higher level of neurons, onto the neuron at r, through the synapses at the lower level, s.

3) Computational Implementation of density and density-connectivity data



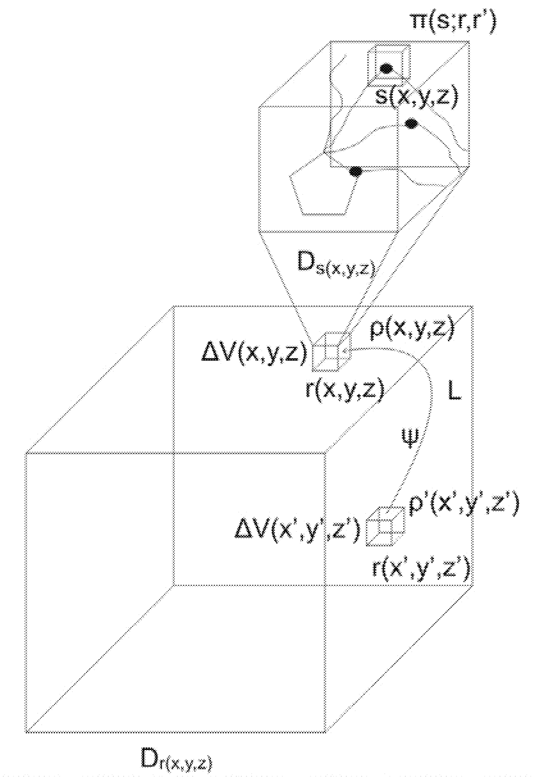


Figure 3: Neuronal connectivity in the physical space *(left)* and it correspondence with abstract mathematical space *(right)*. *Right*: domains $D_r(r)$ of neurons connected with neurons at r, $D_s(r)$ of synapses inside neurons at r connected with neurons at r'. *Left*: Geometrical representation of the domain of neurons $D_{r(x,y,z)}$ with a functional interaction ψ between two neurons at $r(x,y,z)$ and $r(x',y',z')$. Zooming on the volume $\Delta V(x,y,z)$ where the post synaptic neuron is located shows its synaptic domain $D_{s(x,y,z)}$.

Legend:

D_r : Space of neurons; $D_{r(x,y,z)}$ the same space in the physical space (cartesian coordinates)

D_s : Space of synapses; $D_{s(x,y,z)}$ the same space in the physical space (cartesian coordinates)

$r(x,y,z)$: Localization point of the postsynaptic neuron in a part $\Delta V(x,y,z)$ of in the physical space; r is the same point in the space of neurons.

$r(x',y',z')$: Localization point of the postsynaptic neuron in the physical space; r' is the same point in the space of neurons $\Delta V(x',y',z')$.

$s(x,y,z)$: Localization point of the synapse in the physical space; s is in space of synapses.

$\rho(r(x',y',z'))$: Neuronal density at the point $r(x',y',z')$.
$\rho(r(x,y,z))$: Neuronal density at the point $r(x,y,z)$.
$\pi(s;r,r')$: Synaptic density-connectivity at $s(x,y,z)$.
L : Length of the axon between the two neurons.

The computational implementation of the neuronal density ρ and synaptic density connectivity π follows the previous description. The geometrical data are the neurons localization r and r', the synapse localization s and even the axon length between the two neuron because of the finite propagation velocity of ψ. The corresponding XML code will contain elements that describe the anatomy of the system such as:

Neuronal density:
<NEURONS>
<NEURON name="NEURON1" region=" $r(x',y',z')$ " value=" $\rho(r(x',y',z'))$ "/>
<NEURON name="NEURON2" region=" $r(x,y,z)$ " value=" $\rho(r(x,y,z))$ "/>
<NEURONS/>

Synaptic density-connectivity :
<CONNEXIONS>
<CONNECTION post=" $r(x,y,z)$ "
pre=" $r(x',y',z')$ " region=" $s(x,y,z)$ "
value=" $\pi(s;r,r')$ distance ="L"/>
</CONNEXIONS>

IV. APPLICATION TO CEREBELLUM: COMPUTING IMPLEMENTATION

Research in neuroanatomy and neurohistology provides us with: (i) neuronal densities of each neuron type present in the organ; (ii) the number of axons that innervate each neuron (convergence); and (iii) the number of neurons innervated by each axon (divergence).

As said above, the large number of neurons (about 10^{11}) and interconnections (about 10^{15}) in the cerebellum, leads to considering the two types of densities:

- 6 functions for the neuronal densities ρ^i $(1 \leq i \leq 6)$ for the 6 neuronal, i.e., Purkinje cells, Golgi cells, granule cells, basket cells, stellate cells and nuclei cells.
- One π function for the synaptic density-connectivity between neurons.

1) Cerebellum anatomy

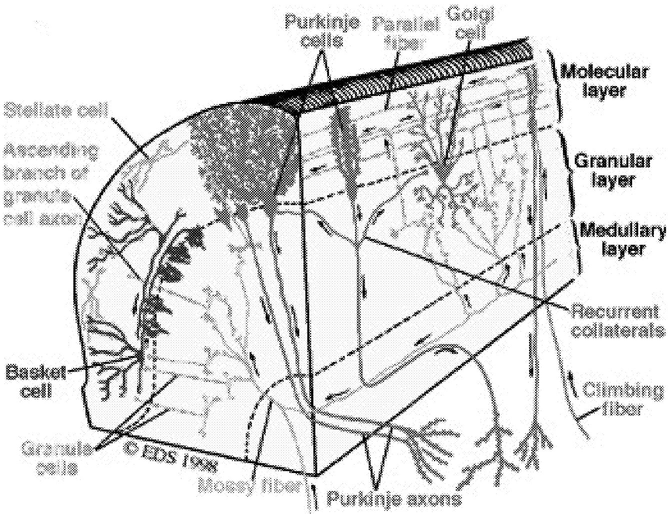


Figure 4: A section of the cerebellar cortex.

As shown in figure 4, the cerebellum is composed of 6 types of neurons:
Purkinje cells, Golgi cells, granule cells, basket cells, stellate cells and nuclei cells that are reached by Purkinje axons.

2) Coordinates reference system

The domain of definition of density functions is the physical space of the cerebellum structure. This structure is very specific and regular. It contains at least one symmetry axis like the sagittal axis for various layers of cells. Then we may use cylindrical coordinates system, where the z-axis is the sagittal axis.

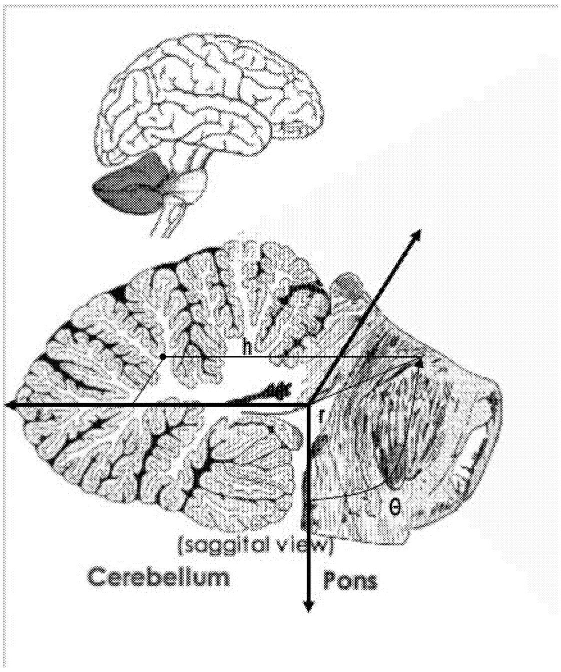


Figure 5: A cerebellum saggital view in a cylindrical coordinate system (http://web.lemoyne.edu/~hevern/)

3) Neuronal density

In the space of neurons D_{r}, we define a density function for each type of the cerebellar neurons, i.e. $\rho^{i}(d,\theta,h)$.

Because of the compartmentalization of the cerebellum in different layers and the assumption that the neuronal densities are constants in a layer, the density functions are assumed to be step functions.

$$D_{r(x,y,z)} \to \mathbb{R}$$
$$(d,\theta,h) \to \rho^{i}(d,\theta,h)$$

4) Synaptic density-connectivity

In the synaptic space D_{s}, for each point s we define synaptic density connectivity between a pre-synaptic neuron in r' and a post-synaptic neuron in r:

$$D_{s(x,y,z)} \times D_{r(x,y,z)} \times D_{r(x',y',z')} \to \mathbb{R}$$
$$(s,r,r') \to \pi(s;r,r')$$

5**) Computational implementation**

In order to be used by the simulator kernel, the anatomical data of the cerebellum will be saved in XML data files.

a- Neuronal density

Because of the compartmentalization of the cerebellar cortex, first we define all the different regions of the cerebellum, and then we define for each region the densities of each type of neuron.

Exemple:

```
<DEFSPACE>
  <SPACE name="tissu" dimension="2"
lengths="2;1">
      <REGION name="PU" inf="0,3;0,2"
sup="1,6;1">
      <REGION name="GR" inf="0,5;0,2"
sup="1,0;0,5"/>
   </SPACE>
</DEFSPACE>
<NEURONS>
         <NEURON name="PURKINJE"
region="PU" value="3,5"/>
         <NEURON name="PURKINJE"
region="GR" value="0"/>
<NEURONS/>
```

b- Synaptic density-connectivity

Because anatomical data are known under the form of neuronal convergence and divergence, an XML file is implemented as:

```
<CONNEXIONS>
         <CONNECTION name="1"
post="purkinje" pre="granular" region="PU"
value="4">
</CONNEXIONS>
```

V. DISCUSSION AND CONCLUSION

Computational modelling of the real nervous network is a difficult task because of the large number of neurons and synaptic connections between them. We have presented in this paper a way to geometrically describe the anatomy of the hierarchical nervous system in the framework of the MTIP, a novel approach for neuronal modeling. This new approach offers many advantages as a general theory for physiological studies using a specific mathematical formalism.

The cerebellum has been chosen as an illustration of the method because of its integration in the sensorimotor system. It is also remarkable because of its simple cytological architecture, highly uniform, with connections organized into a rough, three-dimensional array of perpendicular circuit elements. This organizational uniformity makes the nervous circuitry relatively easy to study. However, we faced some difficulties like the lack of detailed numerical anatomic data, the geometrical representation of the cerebellum in a cylindrical coordinate reference system, the passage from the geometric representation to the formalism.

This geometrical model is a part of the more complete computing system (Physio*Matica*^{TM}) that is used to solve non-local dynamics of physiological functions as learning and memory. More specifically, it will be used as computational input for the simulation of the cerebellum function in order to compare calculated postures with those obtained during development (Jouen et al. 2002).

REFERENCES

Chauvet, G. A. (1993). "Hierarchical functional organization of formal biological systems: a dynamical approach. I. An increase of complexity by self-association increases the domain of stability of a biological system." Phil Trans Roy Soc London B **339**: 425-444.

Chauvet, G. A. (1999). "S-Propagators: a formalism for the hierarchical organization of physiological systems. Application to the nervous and the respiratory systems." Int J General Systems **28**(1): 53-96.

Chauvet, G. A. (2002). "On The Mathematical Integration Of The Nervous Tissue Based On The S-Propagator Formalism. I. Theory." Journal of Integrative Neuroscience **1**(1): 31-68.

Chauvet, P. and G. A. Chauvet (2002). "On the mathematical integration of the nervous tissue based on the S-Propagator formalism. II. Numerical simulations for molecular-dependent activity." J.Integr.Neurosci. **1**(2): 157-194.

Jouen, F., Lejeune, L., Porton-Deterne, I. (2002). Intégration sensori-motrice : les bases de l'intégration posturale. In H. Bloch (ed.). *Les objectifs cognitifs de la prime enfance*. Paris : Hermès Science Publication.

Thompson, R. F. and D. J. Krupa (1994). "Organization of memory traces in the mammalian brain." Annu Rev Neurosci **17**: 519-549.

DEVELOPMENT OF A CARDIOVASCULAR MODEL WITH BARORECEPTOR REFLEX

Jinhuai Lin
Derek G Tilley
Roger F Ngwompo
University of Bath, Claverton Down,
Bath, BA2 7AY,
United Kingdom
Emails: enmjl@bath.ac.uk; D.G.Tilley@bath.ac.uk; R.F.Ngwompo@bath.ac.uk

KEYWORDS

Baroreceptor Reflex; Heart; Human; Mathematics Models; Cardiovascular; physiology; Computer Simultion

ABSTRACT

A mathematical model of the cardiovascular system is described that includes a three-compartment baroreceptor reflex model.

The heart is modelled as a four-chamber time varying elastic balloon and the vascular systems are modelled as elastic and resistant vessels using an exponential equation. The model also includes interventricular septum, pericardium and intrathoracic pressure effects.

The baroreceptor reflex is the most important short-term control mechanisms of the cardiovascular system. This is modelled as a three-compartment model - afferent, central and efferent. The efferent compartment consists of the sympathetic and parasympathetic divisions. These have a direct effect on the heart rate, peripheral resistance and venous compliance.

Predicted responses, including pressure and flow waveforms, show good agreement with published data.

NOTATION

A	Area (cm^2)
a	Coefficient
b	Bernoulli's resistance (mmHg/ (ml/s)^2)
E(t)	Time varying elastance (mmHg/ml)
$\hat{e}$	Effective time varying elastance with septal coupling
E	Elastance (mmHg/ml)
E_0	Elastance constant (mmHg/ml)
Elva	Amplitude component of exponential Elastance (mmHg/ml)
Elvb	Baseline component of exponential Elastance (mmHg/ml)
Es	Elastance of interventricular septum (mmHg/ml)
f	Frequency (Hz)
G	Steady state gain
G^{τ}_{s0}	Steady state gain of the time constant of the sympathetic nerve
HR_0	Heart rate when no sympathetic or vagal stimulation is present
HR_s	Heart rate when only sympathetic stimulation is present
HR_{sv}	Heart rate when sympathetic and vagal stimulation present
HR_v	Heart rate when only vagal stimulation is present
ΔHR	Heart rate change
k_s	Slope at symmetric point of sympathetic sigmoid function
k_p	Slope at symmetric point of parasympathetic sigmoid unction
k_{lr}	Left-to-right ventricular pressure gain
k_{rl}	Right-to-left ventricular pressure gain
K_{pc}	Pericardial pressure coefficient (mmHg)
$k_s^{\tau} / k_{s0}^{\tau}$	Coefficients of the exponential function
L	Inertance (mmHg/ (ml/s^2))
MAP	Mean arterial pressure (mmHg)
P	Pressure (mmHg)
PNA	Parasympathetic nerve activity
Q	Flow (ml)
R	Resistance (mmHg/(ml/s))
SNA	Sympathetic nerve activity
t_{ac}	Time atrium begins to contract (s)
t_{ar}	Time atrium begins to relax (s)
t_{ee}	Time of end-ejection at maximum elv (s)
Tr	Time period of cardiac cycle (s)
V	Blood volume (ml)
Vpc	Heart volume and pericardial fluid volume (ml)
Vpc0	Vpc offset volume (ml)
Φ	Volume constant (ml)
μ	Viscosity of blood
ρ	Density of blood (g/ml)
τ	Time constant
τ lvc	Time constant for left ventricular contraction (s)
τ lvr	Time constant for left ventricular relaxation (s)
Ω	Viscoelastance

Subscript

Aff	Afferent compartment
aa	Aorta
av	Aortic valve
demand	Body demand
error	Error signal
la	Left atrium
lv	Left ventricle
mv	Mitral valve
pc	pericardial
pua	Pulmonary arteries
puc	Pulmonary capillaries
puv	Pulmonary veins
pv	Pulmonary valve
ra	Right atrium
rv	Right ventricle
sa	Systemic arteries
sc	Systemic capillaries
sv	Systemic veins
tv	Tricuspid valve
vc	Vena cava

INTRODUCTION

Mathematical models have been used for many years to evaluate the dynamic response of complex engineering and biological systems, including the human physiological system. One of the earliest digital-computer cardiac, renal and respiratory system models was developed in the 1970s by Dickinson (Dickinson 1972). This was followed by a number of cardiovascular and respiratory models of different complexity (Lo 1995; Melchior et al. 1992; Pennati et al. 2004; Pennati et al. 1997; Sud et al. 1993; Sun et al. 1995). These models tend to concentrate on particular aspects of the physiological system and as such, do not provide a complete representation of the total system.

Tomlinson et al (Tomlinson et al. 1993) developed a simulation model that included the interaction between the respiratory and cardiovascular systems. This model was subsequently used to simulate the interactions between a diver and underwater breathing equipment (Tomlinson et al. 1994). The mechanistic model of the lung used in these studies has recently been changed to a physiological model, that includes damage to the lung, in order to simulate patients in adult intensive care (Tilley et al. 2006). However, these models include a simple non-pulsatile cardiovascular system. One area of improvement to the models is the inclusion of a more complete cardiovascular model.

The mathematical models that have been developed for the human cardiovascular system can vary from very simple ones to multisegment representations of the vascular tree. Generally speaking, the cardiovascular system can be represented as two major blocks. The first is a hydraulic system with a number of distensible vessels in a serial and parallel arrangement with the heart as a pump. The second is a control system that controls the arterial pressure and blood flow.

In the cardiovascular hydraulic system, the heart is the most important component which pumps blood into the pulmonary and systemic arteries. The most important chamber of the heart is the left ventricle (Grood et al. 1974; Suga 1971). Suga (Suga 1971) defined the left ventricle as a 'time-varying elastic model'. This concept was later applied to other chambers of the heart. There is evidence from animal studies to support the use of similar elastance characteristics for the two ventricles (Lausted and Johnson 1999) and it has been shown that the atria can be represented by the time-varying elastance concept. Hill (Grood et al. 1974) introduced a three-parametric model for mechanics, which include contractile, series elastic and parallel elastic elements. Based on this approach, Grood et al (Grood et al. 1974) presented a heart muscle model which successfully predicted the force development during both isometric and isotonic contractions. The arterial bed can be represented as a three-element Windkessel model which consists of inertance, resistance and compliance. Non-linearity can be taken into account by specifying a non-linear compliance (Aversano et al. 1988; Sun 1991; Sun et al. 1995; Sun et al. 1997). Using this model, the effect of gravity or acceleration is easily accounted for by the inclusion of an electromotive force of appropriate strength (Maruyama et al. 1982). The model representing the arterial bed can not readily be applied to the venous segments due to the veins and artery having different features (Sun et al. 1997). Modifications such as venous collapse and valving can be characterized by introducing a non-linearity to the parameters.

The arterial baroreceptors, which refer to aortic and carotid baroreceptors and cardiopulmonary receptors, are the dominant sensors in the control of the cardiovascular system. The most detailed control system model (Leaning et al. 1983) contains separate controls for the heart rate, cardiac contractility, arterial resistances and venous compliance. Each overall relationship between an input and output of the reflex system relies on experimental data correlations. The relationships combine serial and parallel linear and nonlinear functions. These functions are difficult to assign as meaningful physiological terms. The models of a baroreceptor reflex usually contain a series of first order filters, gains and pure time delays which are in parallel. Non-linearity is introduced into the control routine including exponential functions and sigmoid functions (Ursino and Magosso 2003). There are two ways to generate the pulse signal, one of which is Integral Pulse Frequency Modulator (IPFM) (TenVoorde B.J. and Kingma R. 2000). IPFM works such that a pulse is output only when it reaches a certain threshold. Sometimes a high pass filter is used to derive the mean value of arterial pressure over the period of a heart beat so that a pulse output is generated (Ursino 2000).

This paper describes the development of a cardiovascular system model that includes both the hydraulic and the control systems. This has been undertaken using the Bath*fp* dynamic simulation package developed at the University of Bath (Tomlinson et al. 1993). Figure 1 shows the Bath*fp* simulation circuit for the cardiovascular hydraulic system model. It is intended to incorporate this into the human physiological system simulation model being developed at Bath.

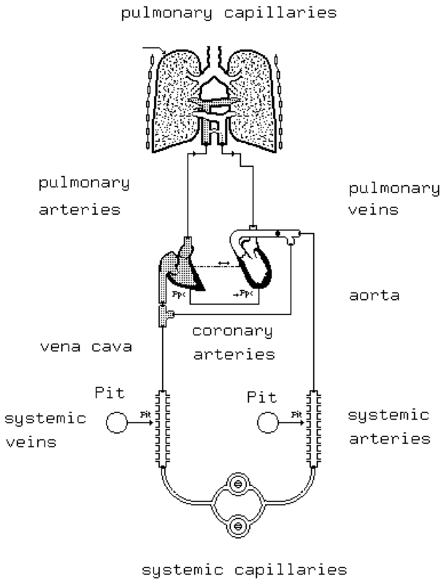


Figure 1: Bath*fp* simulation circuit

MATHEMATICAL MODELS

The cardiovascular hydraulic system

Following inhaled air exchange between the alveoli and blood in the lung capillaries (termed diffusion), blood rich in oxygen is carried to the left heart through the pulmonary veins. The blood passes to the left heart which pumps the blood to the brain, major organs and body tissue via the systemic arteries. The blood flow slows down when it passes through the systemic capillaries because of the large cross-sectional area of the capillaries. Systemic veins serve as low-resistance conduits for venous return to the right heart. The right heart also pumps blood, but in this case the blood flow is into the pulmonary arteries and pulmonary capillaries in the lungs. This is accounted for in the model as follows:

1. The heart is represented as a four-chamber model. As a substantial part of the blood volume entering the ventricle during the diastolic phase is delivered through the atrial contraction (Fung 1984; Zacek and Krause 1996) this effect is also included.

2. Ventricles and atria are modelled as a time-varying elastance model. The elastance term varies over the cardiac cycle according to an exponential charge-discharge waveform and is characterised by a baseline and an amplitude component (Sun et al. 1997). For example, the pressure and volume relationship for the left ventricle is given by

$$P_{lv} = E(t)_{lv} V_{lv} \tag{1}$$

$$E(t)_{lv} = \begin{cases} E_{lva}[1 - e^{-t/\tau_{lvc}}] + E_{lvb} & 0 \le t \le tee \\ (E(tee))_{la} - E_{lvb})e^{-(t-tee)/\tau_{lvr}} + E_{lvb} & tee \le t \le tr \end{cases} \tag{2}$$

3. Interaction between the ventricles is characterised using a three-elastic compartment model. In addition to the elastance of the right and left ventricles, the septum is also represented by an elastic compartment. Under this assumption, the left ventricle (LV) pressure is the sum of the effective LV elastance times volume and the "cross-talk" pressure from the right ventricle (Sun et al. 1997) and is given by

$$P_{lv} = \hat{e}_{lv} V_{lv} + k_{rl} p_{rv} \tag{3}$$

where $k_{rl} = e_{lv}/(E_s + e_{lv})$ and $\hat{e}_{lv} = E_s k_{rl}$.

4. A compartment with an exponential relationship represents pericardial dynamics. The pressure-volume relationship in the pericardium is given by (Sun et al. 1997; Sun and Gewirtz 1988)

$$P_{pc} = K_{pc} e^{(V_{pc} - V_{pc0})/\Phi_{pc}} \tag{4}$$

5. The dynamics of the heart valves are described by Bernoulli's equation, which offers the possibility of investigating the acceleration/deceleration of the flow and the effect of valve stenosis (Fung 1984). For example, the following relationship can be obtained for the mitral valve in the left ventricule

$$L_{mv} \frac{dQ_{mv}}{dt} = P_{la} - P_{lv} - b_{mv} Q_{mv} |Q_{mv}| - R_{mv} Q_{mv} + \Omega_{la} P_{la} \dot{V}_{la} - \Omega_{lv} P_{lv} \dot{V}_{lv} \tag{5}$$

If the friction loss and the Bernoulli term in the equation are neglected, then deceleration is essential for valve closure. For the mitral valve, the flow decelerates after blood injects into the ventricle. The deceleration causes the pressure increase in the direction of flow. When the pressure acting on the ventricular side of the mitral valve becomes higher than that on the other side, the net force acts to close the valve. The same principle applies to the aortic valve. The Bernoulli term in the equation can be regarded as the dynamic pressure while the other pressure term is the static pressure. Hence, pressure gradients act as a driver force to the flow, which contribute to the acceleration/deceleration of the blood flow.

6. The Vascular segments use a four-element model (Sun et al. 1997). Viscoelastance is included besides capacitance, resistance and inertance. Capacitance is assumed to be non-linear and to have an exponential pressure-volume relationship. For example, the pressure-volume relationship of the aortic artery is represented as follows

$$\frac{dQ_{aa}}{dt} = \left[(P_{aa} - P_{sa}) - R_{aa} Q_{aa} + \Omega_{aa} \frac{dV_{aa}}{dt} - \Omega_{sa} \frac{dV_{sa}}{dt} \right] / L_{aa} \tag{6}$$

$$P_{aa} = E_{aa} \Phi_{aa} \tag{7}$$

$$E_{aa} = E_0 e^{V_{aa}/\Phi_{aa}} \tag{8}$$

Physiological characteristics of the veins are different from arteries. Equation (7) and (8) can be used to describe a vein's characteristic with some adjustment. A high value has been assigned to the volume constant Φ as the capacitance of veins is relatively constant and is more compliant than arteries.

7. A respiratory effect on hemodynamics is included by introducing the intrathoracic pressure into the model (denoted as Pit in Figure 1). On the basis of reported data, the intrathoracic pressure varies from -3.7 mmHg during expiration to -5.5 mmHg during inspiration (Sun et al. 1997)

Baroreceptor reflex control system

Baroreceptor reflex is the most important short-term regulator of the arterial pressure and cardiovascular function. If the arterial pressure decreases, for example during a hemorrhage, this causes the discharge rate of the arterial baroreceptors to decrease. Fewer impulses travel up the afferent nerves to the medullary cardiovascular centre, and this induces

- increased heart rate due to the increased sympathetic activity to the heart and decreased parasympathetic activity.
- increased ventricular contractility due to the increase in sympathetic activity to the ventricular myocardium.
- arteriolar constriction due to the increase in sympathetic activity to the arterioles and increased plasma concentrations of chemicals angiotensin II and vasopressin.
- increased venous constriction due to increased sympathetic activity to the veins.

The Baroreceptor reflex is represented as a three-compartment model: afferent, central and efferent components as shown in Figure 2. Aortic pressure is the input parameter to the arterial baroreceptor located in the aortic arch. The baroreceptor then sends the firing pulses through afferent components to the central compartment (central command in Figure 2). The central compartment determines the demands of the body, forming the outflow of the efferent compartment. The efferent compartment consists of sympathetic and parasympathetic nerve divisions. The outputs from the central compartment along the efferent neural pathways terminate at the cell bodies and dendrites of the vagus (parasympathetic) neurons to the heart and the sympathetic neurons to the heart, arterioles, and veins. Although the parasympathetic nerve activity has an impact on heart contractility, the sympathetic nerve activity plays a major role in its control. Therefore, both parasympathetic and sympathetic nerve activity control the heart rate, while other output parameters of the regulator such as heart contractility, venous compliance and peripheral resistance are determined by sympathetic nerve activity.

Ursino and other researchers (Lu et al. 2001; Ursino and Magosso 2003) introduced a non-linearity into the afferent path. Recent research (Kawada et al. 2003), however, shows that the non-linearity should dominate in the efferent pathway rather than at the baroreceptors. Therefore, a non-linear function is used in the two divisions of the efferent nerve flow.

Afferent compartment

The Baroreceptor action potential frequency has a linear relationship with the mean arterial pressure within a pressure range from 40 to 120 mmHg. At a particular steady pressure, there is a certain rate of discharge by the neurons in the baroreceptors. This rate can be increased by raising the arterial pressure, or decreased by lowering the pressure.

A first order filter and a gain element are used to represent the afferent compartment (equation 9) and a saturator is used to limit the pressure within the range from 40 to 120 mmHg.

$$P_{aff} = G_{aff} \frac{MAP}{1+\tau_{aff} s} \tag{9}$$

Central compartment

The primary integrating centre for the baroreceptor reflexes is a diffuse network of highly interconnected neurons called the medullary cardiovascular centre, located in the brainstem medulla oblongata. The neurons in this centre receive inputs from the various baroreceptors. The Central compartment compares inputs with the body demands, and gives out the central point of the sigmoid function as represented by the following equation.

$$P_{error} = P_{aff} - P_{demand} \tag{10}$$

Efferent compartment

The efferent compartment comprises parasympathetic and sympathetic divisions which exhibit a sigmoid shape between the pressure and the nerve activities (Kollai and Koizumi 1989)

$$SNA = \frac{SNA_{\max} - SNA_{\min} e^{Perror/(-ks)}}{1+e^{Perror/(-ks)}} \tag{11}$$

$$PNA = \frac{PNA_{\min} - PNA_{\max} e^{Perror/kp}}{1+e^{Perror/kp}} \tag{12}$$

where k_p and k_s are the slope at the symmetric point of the sigmoid function.

Control of heart rate

Warner (Warner and Russell 1969) investigated the effects of sympathetic and parasympathetic stimulation and the combined effect on the heart rate by stimulating both sympathetic and parasympathetic nerves to a particular frequency level. Three features were drawn from the experiments:

1. the heart rate rises faster than it falls back to the control level.
2. the change in the heart rate due to sudden nerve stimulation is dependent on the frequency stimulated (this is more noticeable for sympathetic than parasympathetic nerve activity)
3. the gain of the response is nonlinear.

A variable-time-constant first order transfer function is included in the path of both nerve divisions at the effector sites to represent the different heart rate response to increment and decrement of the nerve frequency. An exponential relationship with the input nerve frequency is used for the sympathetic branch time constant. A second order polynomial function is used to represent the nonlinear steady state gain (Levy and Zieske 1969). However, the steady state gain is only valid when the stimulated frequency is less than 8 Hz. According to experimental data of Levy and Warner, an exponential function is used to limit the gain. Warner also discovered a certain time delay for both nerve divisions. Therefore, a pure time delay is included in both divisions. Figure 3 and Figure 4 show the predicted response

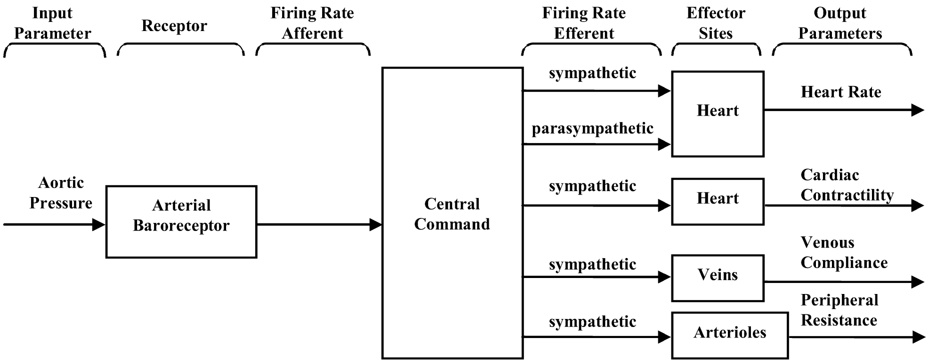


Figure 2: Baroreceptor reflex control flow chart

of the heart rate to sympathetic and parasympathetic stimulation compared with the experimental data from Warner's experiments.

Interactions between the two nerve divisions on the control of the heart rate are taken into account. It is believed that the parasympathetic nerve is dominant over the resting period. Levy et al (Levy and Zieske 1969) interpret this interaction in the following form

$$\begin{aligned}\Delta HR = a_1 \cdot SNA + a_2 \cdot SNA^2 + a_3 \cdot PNA + a_4 \cdot PNA^2 \\ + a_5 \cdot SNA \cdot PNA\end{aligned} \qquad (13)$$

where a_1 to a_5 are coefficients obtained from experimental data fitting.

The other interpretation is from Warner (Warner and Russell 1969)

$$HR_{sv} = HR_v + (HR_s - HR_0)\frac{(HR_v - HR_{\min})}{(HR_0 - HR_{\min})} \qquad (14)$$

These two mathematical equations have been combined to provide a more comprehensive form as follows

$$HR = HR_0 + \Delta HR \qquad (15)$$

$$\Delta HR = HR_s + HR_v + c \cdot HR_s \cdot HR_v \qquad (16)$$

$$HR_s = G_s \frac{f_s \cdot e^{-T_s s}}{1 + \tau_s s}, \qquad (17)$$

where $G_s = G_{s0}(1 - e^{-ks \cdot fs})$ $\qquad k_s = G_{s0} / k_{s0}$

$\tau_s = G_{s0}^{\tau}(1 - e^{-k_s^{\tau} fs})$ $\qquad k_s^{\tau} = G_{s0}^{\tau} / k_{s0}^{\tau}$

$$HR_v = G_v \frac{f_v \cdot e^{-T_v s}}{1 + \tau_v s}, \qquad (18)$$

where $G_v = G_{v0}(1 - e^{-kv \cdot fv})$, $k_v = G_{v0} / k_{v0}$, $\tau_v = Const.$

Other effector sites

Other effector sites including arterioles and veins share similar characteristics. Sympathetic nerve outflow sends the signal to the effector sites which includes a first order filter and a pure time delay as shown in equation (19).

$$\sigma = G_{eff} \frac{SNA \cdot e^{-T_\sigma s}}{1 + \tau_\sigma s} \qquad (19)$$

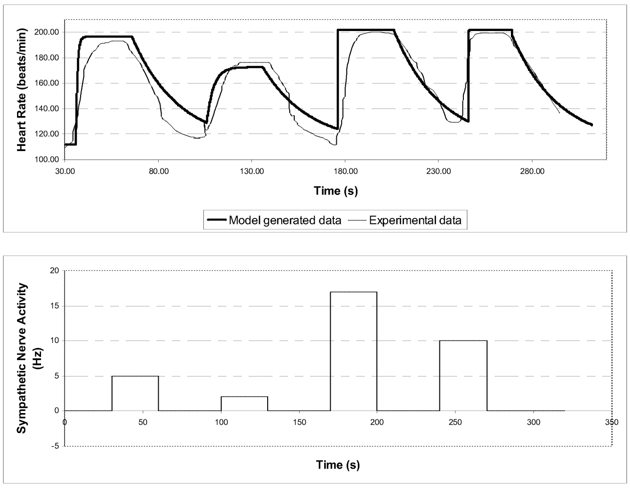


Figure 3: Heart rate response to the sympathetic nerve stimulations. Experimental data adapted from Warner and Cox 1962

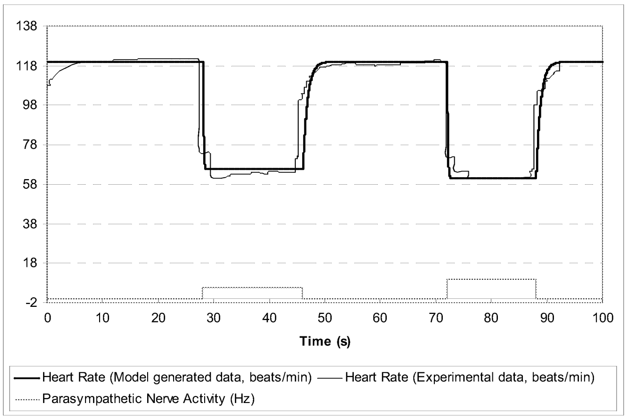


Figure 4: Heart rate response to parasympathetic nerve stimulations. Experimental data adapted from Warner and Cox 1962

RESULTS

The primary cardiovascular model contains 10 components. Parameter values of the integrated model were tuned so as to represent a realistic cardiovascular system. Two criteria were used to justify the results: an accurate representation of the baseline hemodynamics in various parts of the system and a good fit to published pressure and flow waveforms. Figure 5 shows the predicted pressure waveforms which agree with measured data. Simulated pressures and volumes are consistent with in vivo data (Table 1). The trend of decreasing pressure from the aorta to veins agrees with normal physiology. Important hemodynamic indexes such as cardiac output, end diastole and end systole pressures, mean volumes and flows are also in agreement with published data. Sensitivity analysis shows that the model is insensitive to most of the parameters. Pericardial volume offset is the most sensitive parameter mainly because the constraint effect of the pericardium is dominant.

Predicted aortic flow exhibits similar waveforms to experimental data, due to the use of the nonlinear characteristics for arteries in this model (Figure 6). The phenomena including reported backflow and noise during the pressure decreasing phase for the aortic and mitral valves are neglected since such detailed modelling is beyond the scope of the current work.

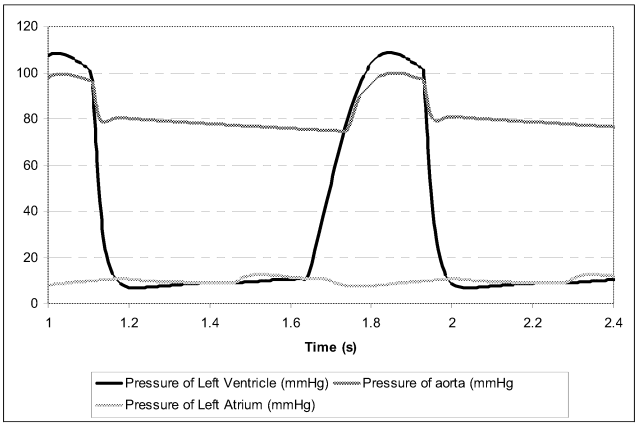


Figure 5: Model simulated pressure of left ventricle, aorta and left atrium

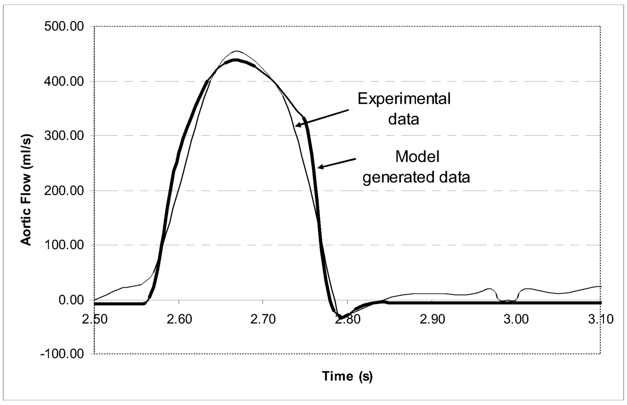


Figure 6: Aortic flows. (Experimental data adapted from Sugawara et al. 1989)

Table 1: Model representation of pressures and volumes compared with in vivo data* assuming the body surface is 1.63 m^2

		Data from Fung 1984 and Berne and Levy 1988	Model representation
Left atrial pressure (mean)		≤ 12 mmHg	8.00 mmHg
Left ventricular pressure	Peak systolic	100~150 mmHg	107.7 mmHg
	End - diastolic	≤ 12 mmHg	4.60 mmHg
Aortic pressure	Systolic	100~150 mmHg	99.90 mmHg
	Diastolic	60~100 mmHg	74.9 mmHg
Right atrial pressure (mean)		≤ 6 mmHg	5.95 mmHg
Right ventricular pressure	Peak systolic	15~30 mmHg	26.1 mmHg
	End - diastolic	≤ 6 mmHg	3.10 mmHg
Pulmonary arterial pressure	Systolic	15~30 mmHg	17.30 mmHg
	End - diastolic	4~12 mmHg	10.50 mmHg
Stroke volume*		65~114 ml	70.4 ml
Pressure of pulmonary capillaries (mean)		10 mmHg	14.6 mmHg
Pressure of pulmonary veins (mean)		6 mmHg	8.87 mmHg
Systemic capillaries (mean)		20 mmHg	21.67 mmHg
Systemic veins (mean)		10 mmHg	7.21 mmHg

Figure 7 includes the predicted and measured blood flow rates for the mitrial valve. The shaded areas indicate the contribution due to the contraction of the atrium, which normally contributes around 15% of the ventricular filling. This activity becomes more important when the heart rate increases. However, an appropriately timed atrial contraction is not necessary for complete end-diastolic mitral valve closure. Clinical data shows that the mitral valve closes completely in the absence of an atrial contraction. Therefore, atrial contraction mainly contributes to ventricular filling and not to the valve closure.

The blood vessels are represented by capacitance, resistance, inertance and viscoelastance terms. Conventionally, the capacitance or compliance is represnted using a linear function that depends on the volume and pressure of the vessels. However, experiments show that the arteries become stiffer in the high-pressure range (Melchior et al. 1992), indicating at the compliance of the arteries varies with pressure. An exponential pressure-volume equation included in the model effectively reprecates this behaviour without introducing a discontinuity into the model (Figure 8).

Interactions exist everywhere in a complex biological system such as the cardiovascular system. For example, interactions between the ventricles, including pressure and volume coupling, affect the performance of the heart significantly. The elastance of the interventricular septum causes the volume of left ventricle to interact with the right ventricle. Pericardial coupling restrains the heart chamber from over distension. Jackini and Weber (Janicki and Weber 1980) suggested that the pericardium will increase end diastole pressure by 4.6 mmHg while a much larger increase of end systole pressure was reported by other authors. This phenomenon can be assessed in the model by setting the gain Kpc in equation 4 to zero. The resulting shift in the pressure and volume waveforms are shown in Figure 9.

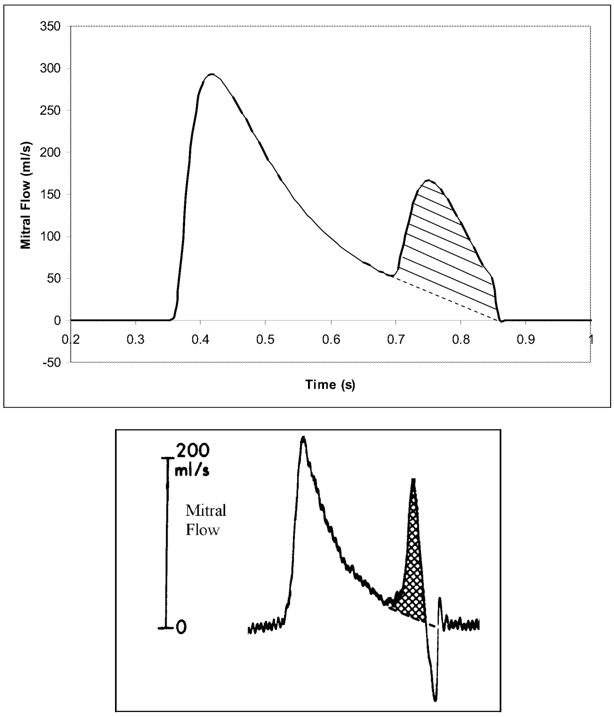


Figure 7: Mitral valve flow. Upper panel: model generated data; Lower panel: experimental data (Sugawara et al. 1989). Shaded area due to atrial contration

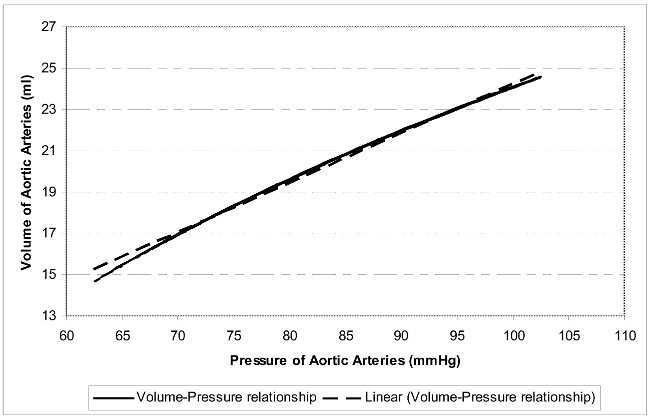


Figure 8 Volume – Pressure relationship of aortic arteries

Respiratory effects on the cardiovascular system have also been considered. The Valsalva manoeuvre is a useful bedside test of autonomic function. By applying an intrathoracic pressure difference between the systemic and pulmonary circulations, the model is able to demonstrate the manoeuvre as shown by the results presented in Figure 10 and Figure 11.

During expiration, the intrathoracic pressure is elevated. This leads to the holding back of venous return and the pulmonary reservoir is gradually depleted. And the arterial pressure decrease below the normal level. Without autonomic control, the blood pressure falls and remains low until the intrathoracic pressure is released. For this condition, the heart rate remains constant at 70 beats/min as shown in Figure 10. If the baroreceptor reflex is activated, it causes vasoconstriction and a tachycardia, to raise blood pressure. This is shown in Figure 11 where the heart rate increases to over 100 beats/min. For inspiration, the arterial pressure increases to the normal level as a result of the retained venous return. The restore of venous return causes an overshoot of the blood pressure, which leads to a baroreceptor mediated bradycardia when the baroreceptor reflex is activated (Figure 11). This overshoot is absent without autonomic control (Figure 10).

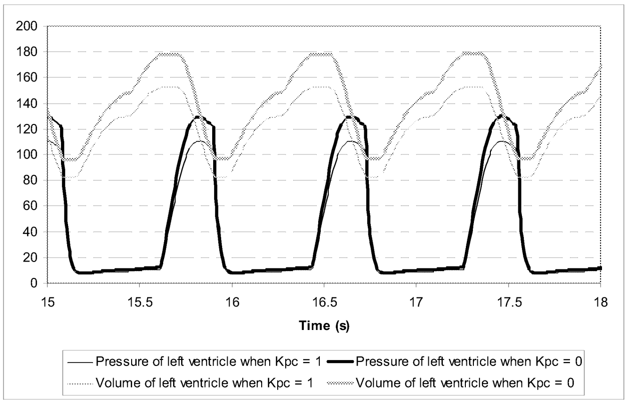


Figure 9: Pericardium effect on ventricles pressure and volume

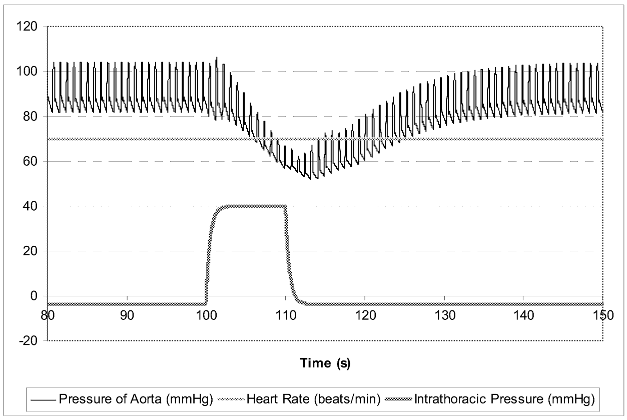


Figure 10: Valsalva manoeuvre without baroreceptor reflex

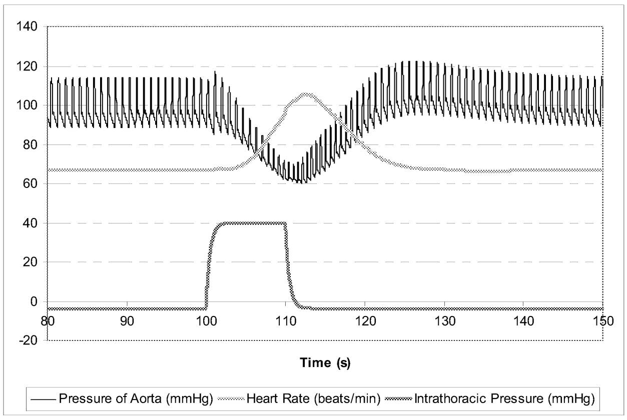


Figure 11: Valsalva manoeuvre with baroreceptor reflex

CONCLUSIONS

A mathematical model of the cardiovascular system is described, based on a three-compartment baroreceptor reflex model. This model has ten elements including pulmonary and systemic capillaries as resistant vessels. The model processes including decomposition, parameter estimation, sensitivity analysis and heuristic tests are discussed. The results obtained from the model are in good agreement with published data. The model is able to predict characteristic changes in pressure and flow waveforms as direct hemodynamic consequences. Interactions including pericardium, interventricular septum and respiratory effects are taken into account. A Valsalva manoeuvre is used to demonstrate the respiratory effects on the cardiovascular system. Further optimisation is needed for cardiovascular system parameters and more case studies are required to fully validate the model.

REFERENCES

Aversano,T., W.L.Maughan, K.Sunagawa, and L.C.Becker. 1988. "Effect of Afterload Resistance on End-Systolic Pressure-Thickness Relationship." *American Journal of Physiology*. 254:H658-H663.

Berne,R.M. and M.N.Levy. 1988. *Physiology*. The C.V.Mosby Company. the USA.

Dickinson,C.J. 1972. "Digital-Computer Model to Teach and Study Gas Transport and Exchange Between Lungs, Blood and Tissues(Macpuf)." *Journal of Physiology-London*. 224:7-&.

Fung,Y.C. 1984. *Biodynamics: Circulation*. Springer-Verlag. New York.

Grood,E.S., R.E.Mates, and H.Falsetti. 1974. "Model of Cardiac-Muscle Dynamics." *Circulation Research*. 35:184-196.

Janicki,J.S. and K.T.Weber. 1980. "The Pericardium and Ventricular Interaction, Distensibility, and Function." *American Journal of Physiology*. 238:H494-H503.

Kollai,M. and K.Koizumi. 1989. "Cardiac Vagal and Sympathetic-Nerve Responses to Baroreceptor Stimulation in the Dog." *Pflugers Archiv-European Journal of Physiology*. 413:365-371.

Lausted,C.G. and A.T.Johnson. 1999. "Respiratory resistance measured by an airflow perturbation device." *Physiol Meas.* 20:21-35.

Leaning,M.S., H.E.Pullen, E.R.Carson, and L.Finkelstein. 1983. "Modelling a complex biological system: the human cardiovascular system. 2. Model validation reduction and development." *Transactions on Instrumentation, Measurement, and Control*. 5:87-98.

Levy,M.N. and H.Zieske. 1969. "Autonomic control of cardiac pacemaker activity and atrioventricular transmission." *Journal of Applied Physiology*. 27:465-470.

Lo,J.K.W. Mathematical modelling of mixed gas breathing equipment and associated system. 1995. University of Bath.
Ref Type: Thesis/Dissertation

Maruyama,Y., K.Ashikawa, S.Isoyama, H.Kanatsuka, E.Inooka, and T.Takishima. 1982. "Mechanical Interactions Between 4 Heart Chambers with and Without the Pericardium in Canine Hearts." *Circulation Research*. 50:86-100.

Melchior,F.M., R.S.Srinivasan, and J.B.Charles. 1992. "Mathematical modeling of human cardiovascular system for simulation of orthostatic response." *Am J Physiol.* 262:H1920-H1933.

Pennati,G., M.Bellotti, and R.Fumero. 1997. "Mathematical modelling of the human foetal cardiovascular system based on Doppler ultrasound data." *Med.Eng Phys.* 19:327-335.

Pennati,G., G.B.Fiore, K.Lagana, and R.Fumero. 2004. "Mathematical modeling of fluid dynamics in pulsatile cardiopulmonary bypass." *Artif.Organs*. 28:196-209.

Sud,V.K., R.S.Srinivasan, J.B.Charles, and M.W.Bungo. 1993. "Mathematical modelling of the human cardiovascular system in the presence of stenosis." *Phys.Med.Biol.* 38:369-378.

Suga,H. 1971. "Theoretical Analysis of A Left-Ventricular Pumping Model Based on Systolic Time-Varying Pressure/Volume Ratio." *Ieee Transactions on Biomedical Engineering*. BM18:47-&.

Sugawara,M., R.D.Kajiya, A.Kitabatake, and H.Matasuo. 1989. *Blood Flow in the heart and Large Vessels*. Springer-Verlag. Tokyo.

Sun,Y. 1991. "Modeling the dynamic interaction between left ventricle and intra-aortic balloon pump." *Am J Physiol.* 261:H1300-H1311.

Sun,Y., M.Beshara, R.J.Lucariello, and S.A.Chiaramida. 1997. "A comprehensive model for right-left heart interaction under the influence of pericardium and baroreflex." *Am.J.Physiol.* 272:H1499-H1515.

Sun,Y. and H.Gewirtz. 1988. "Estimation of intramyocardial pressure and coronary blood flow distribution." *Am J Physiol.* 255:H664-H672.

Sun,Y., B.J.Sjoberg, P.Ask, D.Loyd, and B.Wranne. 1995. "Mathematical model that characterizes transmitral and pulmonary venous flow velocity patterns." *Am J Physiol.* 268:H476-H489.

TenVoorde B.J. and Kingma R. 2000. "A baroreflex model of short term blood pressure and heart rate variability." *Stud Health Technol Inform*. 71:179-200.

Tilley,D.G., A.W.Miles, C.M.Murphy, B.S.Brook, A.J.Wilson, and D.Breen. Computer Modelling of the Human Respiratory system in Adult Intensive Care,I-Mathematical Modelling. 2006.
Ref Type: Unpublished Work

Tomlinson,S.P., J.Lo, and D.G.Tilley. 1993. "Time Transient Gas-Exchange in the Respiratory System." *Ieee Engineering in Medicine and Biology Magazine*. 12:64-70.

Tomlinson,S.P., J.K.W.Lo, and D.G.Tilley. 1994. "Computer simulation of human interaction with underwater breathing equipment." *Proceedings of the Institution of Mechanical Engineers, Part H: Journal of Engineering in Medicine*. 208:249-261.

Ursino,M. 2000. "Modelling the interaction among several mechanisms in the short-term arterial pressure control." *Stud Health Technol Inform*. 71:139-161.

Ursino,M. and E.Magosso. 2003. "Role of short-term cardiovascular regulation in heart period variability: a modeling study." *American Journal of Physiology-Heart and Circulatory Physiology*. 284:H1479-H1493.

Warner,H.R. and A.Cox. 1962. "A mathematical model of heart rate control by sympathetic and vagus efferent information." *Journal of Applied Physiology*. 17:349-355.

Warner,H.R. and R.I.C.H.Russell. 1969. "Effect of Combined Sympathetic and Vagal Stimulation on Heart Rate in the Dog." *Circulation Research*. 24:567-573.

Zacek,M. and E.Krause. 1996. "Numerical simulation of the blood flow in the human cardiovascular system." *J Biomech.* 29:13-20.

SIMULATION DYNAMICS IN ECOLOGY AND BIOLOGY

Micro-Gen: An Agent-Based Model of Bacteria-Antibiotic Interactions in Batch Culture.

James T. Murphy, Ray Walshe
Biocomputation Research Laboratory,
School of Computing,
Dublin City University,
Dublin 9,
Ireland.
E-mail: {James.Murphy, Ray.Walshe}@computing.dcu.ie

KEYWORDS

Biology, Simulator, Object-oriented, Health sciences.

ABSTRACT

The software model 'Micro-Gen' simulates the growth and development of cultured bacterial cells and their interactions with antimicrobial drugs. It uses an agent-based modelling approach, which means that pre-defined rules and parameters for each bacterium can be set, and emergent properties of the colony examined, without the need for population-level laws. The model reproduces accurately the standard growth curve characteristic of bacterial colonies grown in batch culture in the laboratory. It also successfully reproduces the effects of bacteriostatic and bactericidal antibiotics on the growth of the bacterial colony. The model provides a robust platform for incorporating knowledge of the cellular mechanics of individual bacterial cells into an overall model of heterogeneous bacterial population dynamics.

INTRODUCTION

In recent years, the development of antibiotic resistance in bacteria has become an important health risk that poses a continuing threat to the viability of many forms of antibiotic treatment (Andersson 2003). The question of how resistance develops and spreads within populations of bacteria is of crucial importance for implementing effective treatment regimes. Computational analysis of bacterial population dynamics provides a logical basis for investigations into the effects of exposure to different types of antibiotics, and other environmental factors, on bacterial growth and development. This complements molecular and cellular based studies, by facilitating an integrated perspective for investigating the key components and interactions at work in the system.

There are a number of different approaches possible for modelling the growth of bacterial colonies. Mathematical population models are commonly used for describing the growth and development of the colony as a unit, using global parameters (Lacasta et al. 1999). Another common method is based on cellular automata theory, which has been used previously to explain pattern formation in colonies (Ben-Jacob et al. 1994). A third approach is the agent-based (or individual-based) modelling approach used here (Ginovart, Lopez and Valls 2002; Kreft, Booth and Wimpenny 1998). The most significant characteristic of the agent-based approach is that the biological properties of the individual bacterial cells are used to determine the progress of the simulation. This allows a finer-grained analysis, correlating local changes at the individual level to global effects at the population-level.

The software model 'Micro-Gen' (which stands for 'Microbial Genetics'), described here, was developed to simulate the growth and development of cultured bacterial cells and their interactions with antimicrobial drugs. It uses an agent-based modelling approach to simulate the interactions of individual bacterial cells and antibiotic molecules in a discrete two-dimensional environment. This account for spatial heterogeneity of nutrient content and the distribution of bacteria, as opposed to assuming a completely homogeneous, mixed environment. Pre-defined rules and parameters for each bacterium can be set and emergent properties of the population examined, without the need for global (population-level) laws (Jennings, Sycara and Wooldridge 1998).

DESCRIPTION OF THE MODEL

Overview

Micro-Gen is coded in the C++ object-oriented programming language. It is a modified and extended version of BAIT (Bacterial-Antibiotic Interaction Tool), a Java software tool for individual-based modelling of bacterial growth, previously developed in this lab (Walshe 2006). The bacterial agents occupy a discrete, 2-dimensional lattice environment consisting of a grid of 'Patches' which each contain a specified number of nutrient molecule objects. The bacteria obtain energy by "feeding" off of a free nutrient molecule present in the same patch. Only one bacterium may occupy a nutrient molecule at any given time, and since there are only a limited number of molecules per patch, this restricts the number of bacteria that may occupy a single patch at the same time. Each loop, or time-step, of the simulation represents 2.8 seconds in real-time, corresponding to an optimal bacterial doubling time of ~20 minutes (Walshe 2006).

The program includes significantly refined and improved behavioural rules for the agents compared to the original BAIT software. The model of the bacteria's environment has also been modified so that either a liquid batch culture or a solid agar medium can be simulated. Significant performance enhancements and memory optimizations have been carried out in order to allow the model to be scaled up to accommodate substantially larger numbers of agents and to reduce the processing requirements.

Bacteria Agents

Initially, when the bacteria are added to the medium, their rate of nutrient uptake is at a base level which gradually increases to the optimal rate of uptake after a specified interval. This represents the lag stage of growth when the bacteria have not yet adapted to their environment, and a specified period of time is required to synthesise the enzymes required for nutrient uptake (Swinnen et al. 2004). As the bacteria consumes nutrient molecules during its life cycle, it increases its internal store (or stock) of nutrient. When the bacterium's nutrient stock reaches a specified maximum level, reproduction occurs by the process of binary fission. The cell divides into two cells, each with approximately half the original nutrient stock. Under optimal nutrient conditions this results in an exponential phase of growth. The conditions for optimum growth depend on factors such as temperature, pH and nutrient availability (Ross et al. 2003).

During each time step of the simulation, a survival cost is subtracted from the cell's nutrient stock, and there is also a movement cost associated with the energy expended in propulsion. If the cell's nutrient stock drops below a specified level, the bacterium enters the stationary phase during which it attempts to conserve energy (Nystrom 2001, Siegele, Kolter 1992). The survival cost is reduced to represent the decreased metabolic rate characteristic of stationary phase bacteria, and the bacterium ceases propelled locomotion. When the nutrient stock eventually drops to zero, cell lysis occurs and the bacterium dies, releasing a trace amount of nutrient back into the medium.

The bacterial agents in Micro-Gen implement the "run and tumble" mode of movement characteristic of species with flagella (Mitchell, Kogure 2006). They move forward smoothly during the "run" phases and reorient to a random direction during the alternating "tumble" phases. They also display positive chemotaxis when exposed to nutrient concentration gradients. This is facilitated by a temporal sensing system whereby the bacteria periodically compare nutrient concentrations as they move from patch to patch through the environment (Segall, Block and Berg 1986). When a bacterium encounters a positive nutrient gradient it lengthens the time of its "run" phase, and the relative duration of runs and tumbles determines if the cell moves towards or away from a chemical environment.

Antibiotic Agents

Antibiotics are represented as software agents with rules dictating their movement and interactions with bacteria. The antibiotic molecules diffuse through the medium by a modified Random Walk algorithm. The probability of a molecule moving to an adjacent patch is proportional to the concentration of molecules present in the current patch. It is based on the principal that diffusion occurs by the random collision of molecules as they undergo Brownian motion.

When an antibiotic occupies the same patch as a bacterium they will interact based on the rules assigned for antibiotic-bacteria interactions. A cumulative antibiotic damage coefficient is associated with each bacterium and every time it interacts with an antibiotic this is incremented. When the damage coefficient reaches a specified level, growth of the cell is inhibited (bacteriostatic effect). If the antibiotic is bactericidal, then further interactions will result in increases of the damage coefficient, until death occurs when a specified level is reached.

The user can specify whether the antibiotic agent is destroyed after interacting with the bacterium, or else remains "active" for the duration of the simulation. The latter option represents situations where the concentration of anitibiotic available is sufficiently high so as not to be a limiting factor over the time course of the experiment. Otherwise, depletion of antibiotic over time by degradation/metabolisation occurs.

Nutrient Diffusion

The diffusion of nutrient in the environment is applied using a modified implementation of Fick's law of diffusion for a discrete, 2-dimensional domain (Ginovart et al. 2002). The difference in nutrient levels between two adjacent patches is calculated and multiplied by the diffusion rate parameter to determine the amount of nutrient exchanged. For diagonally adjacent patches, the diffusion rate is modified by the factor $1 / \sqrt{2}$ to account for the angle at which the patches adjoin.

RESULTS AND DISCUSSION

The rules dictating the behaviour and interactions of the software agents are based on experimentally derived insights on bacterial cell biology drawn from the scientific literature. Parameters applicable to the bacterial species *Escherichia coli* were chosen for a case study simulation using the Micro-Gen software modelling tool. Figure 1 displays the growth curve resulting from a simulation of *E. coli* growth in batch culture using the Micro-Gen software tool. The model reproduces accurately the standard growth curve characteristic of bacterial colonies grown in batch culture in the laboratory.

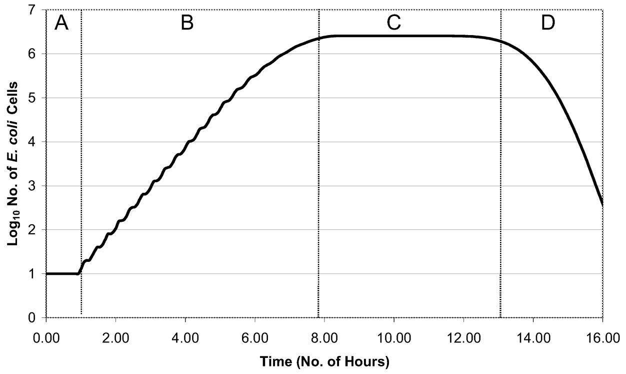


Figure 1: Bacterial Growth Curve Produced by Micro-Gen Software Model. The Four Standard Phases of Growth are Labelled A-D

This global growth pattern is an emergent property from rules assigned at the individual level based on current knowledge of bacterial physiology. The standard growth curve consists of four main stages: the [A] lag, [B] logarithm (or exponential), [C] stationary and [D] death phases.

The lag phase [A] is the initial period after inoculation in fresh culture medium when cell division has not begun to occur (Swinnen et al. 2004). During this phase, the bacteria begin to synthesize macromolecules required to transport and process the nutrients from their new environment. Once they have adapted to their environment, cell division begins to occur and the bacteria enter the logarithmic phase [B] of growth. This is typified by an exponential rate of increase in cell numbers until nutrient availability or accumulation of waste products begins to limit growth.

When the nutrient content of the medium has been exhausted, the bacteria typically enter the stationary phase [C] where no net increase/decrease in cell numbers is observed. Bacteria in this phase are characterised by a metabolically less active and more resistant state (Nystrom 2001, Siegele, Kolter 1992). A low level of endogenous metabolism is maintained and the rate of protein turnover by the cell increases. However, as nutrient starvation persists, eventually most of the cells enter the death phase [D], characterised by an exponential decrease in viable cell counts.

Tests on the effects of bacteriostatic or bactericidal antibiotics with varying concentrations and degrees of susceptibility were carried out. Figure 2 displays results from exposure of an *E. coli* colony to either bacteriostatic or bactericidal antibiotics. The antibiotic concentrations were maintained at a constant level over the time course of the simulation. Bacteriostatic antibiotics inhibit growth and division of the bacterial cells, but are not lethal at clinically relevant concentrations. As a result, growth of the colony is inhibited and the growth curve levels off, as seen in the graph below. On the other hand, bactericidal antibiotics are capable of killing the cells at clinically relevant doses, thus causing an observable decrease in the number of bacteria over time.

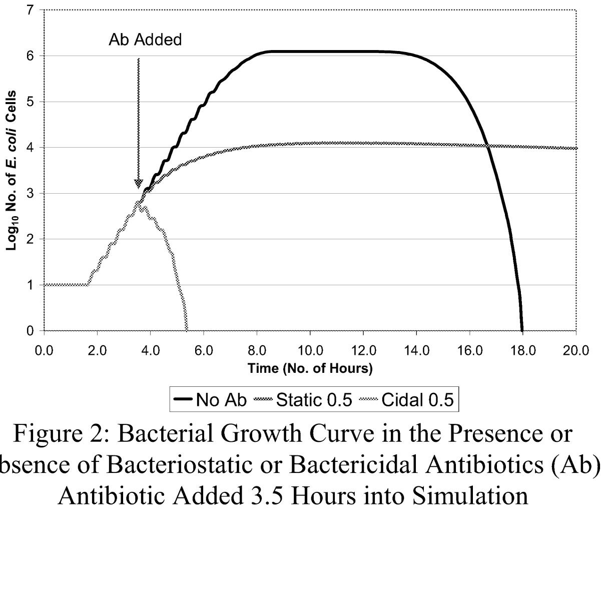


Figure 2: Bacterial Growth Curve in the Presence or Absence of Bacteriostatic or Bactericidal Antibiotics (Ab). Antibiotic Added 3.5 Hours into Simulation

The impact of positive chemotaxis by the bacterial agents in the presence of nutrient gradients was also analysed. The bacteria possess a temporal sensing system to detect nutrient gradients and move towards areas of higher nutrient concentration. Figure 3 displays results from a test where two localised areas with higher nutrient concentrations were created in the top-left and bottom-right corners of the environment. As the nutrient diffuses into the rest of the environment it produces a gradient and the bacteria respond by moving up this gradient. In Figure 3 (taken at the 2 hour time point), the bacterial cells, marked in red, can be seen clustered around the areas of high nutrient concentration, represented by lighter shades of grey. The bacteria were initially randomly distributed across the environment at the beginning of the simulation.

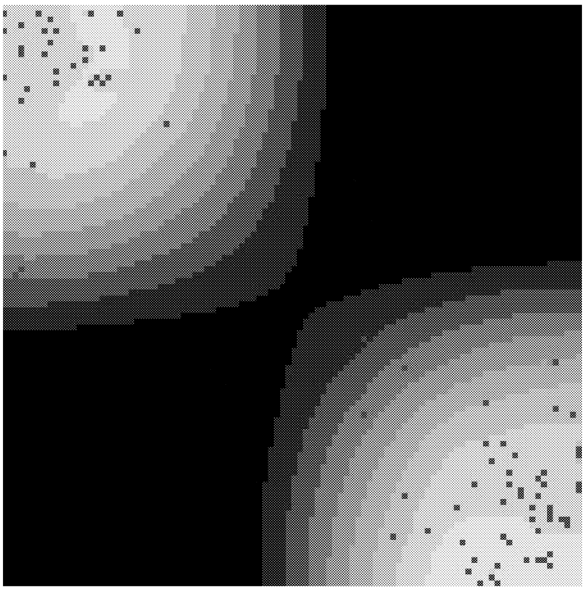


Figure 3: Demonstration of Positive Chemotaxis of Bacteria towards Nutrient Sources. Red Dots = Bacteria. Shades of Grey = Nutrient Concentrations (Lighter Shade = Higher Nutrient Concentration). Simulated Time = 2 hours.

Under the same conditions and parameters, Micro-Gen processes loops of the simulation over 10^2 times faster than the original BAIT software. As a result of the performance and memory optimizations, simulations involving $>3.0\text{x}10^6$ bacterial agents can be carried out using a standard desktop Pentium IV PC with 1Gb of memory. Since the emergent properties of the bacterial colony are affected by the individual interactions and variability of the agents, the number of agents used is a key factor in the realism of the model.

In order to resemble laboratory conditions, where colonies consisting of $>10^7$ individual bacteria per ml of culture can be present, the model has been designed to be able to scale to similar orders of magnitude. Future work will involve parallelisation of the model to take advantage of multi-processor machines with increased memory capacities. The bacterial environment is currently represented in two spatial dimensions, due to computational constraints. However, with greater computing power available, it could be extended to three dimensions.

Micro-Gen currently uses by default the pseudo random number generator from the C++ standard library to introduce variability into the model. However, this is insufficient for more complex simulations due to its short period (2^{31}). Therefore, more sophisticated pseudo random number generators with longer periods, such as the Mersenne Twister, will be incorporated into upcoming versions of the simulator (Matsumoto and Nishimura 1998).

One of the challenges of biological modelling is to find sufficient quantitative biological data to develop accurate rules and parameters for the system. In the case of agent-based models this problem is particularly acute since the majority of experimental data collated for bacteria has consisted of population-averaged results. In this case, the results would be similar to mathematical models using global-level parameters. However, the agent-based approach is advantageous in that parameters for individual bacteria, such as growth rate or antibiotic susceptibility, can be varied and the impact of spatial heterogeneity in the environment taken into account.

Recently, a greater emphasis has begun to be put on collecting experimental information about the life cycles of individual bacterial cells (Elfwing et al. 2004; Niven et al. 2006). The development of new techniques to study individual cells will allow more advanced models to be developed and correlated with existing data. In return, the individual-based modelling approach of Micro-Gen provides an intuitive way of incorporating this knowledge of the cellular mechanics into an overall model of heterogeneous bacterial population dynamics.

Future work on Micro-Gen will include the development of a genetic component to the bacterial agents to simulate the evolution of antibiotic resistance. A preliminary genetic component, representing a DNA sequence which is subjected to random nucleotide substitutions during replication, is already present in the bacterial agents as a proof-of-concept feature. The aim is to extend Micro-Gen to include a detailed model of antibiotic resistance development and evolution in response to antibiotic exposure.

ACKNOWLEDGEMENTS

The authors wish to acknowledge the contribution of Mathieu Joubert in developing a C++ version of the BAIT software tool. We would also like to acknowledge the collaboration of Dr. Marc Devocelle (Royal College of Surgeons in Ireland, Dublin, Ireland) on this project.

AUTHOR BIOGRAPHY

JAMES T. MURPHY graduated from University College Dublin, Ireland, in 2002 with a Bachelor's degree in Cell & Molecular Biology. He completed a Master's Degree in Bioinformatics at Dublin City University in 2004, and is currently in the second year of a Ph.D degree at the School of Computing in Dublin City University. His project involves investigating agent-based approaches to modelling antibiotic resistance frequencies in pathogenic bacteria.

REFERENCES

Andersson, D.I. 2003. "Persistence of antibiotic resistant bacteria". *Current opinion in microbiology,* vol. 6, no. 5, 452-456.

Ben-Jacob, E.; O. Schochet; A. Tenenbaum; I. Cohen; A. Czirok; and T. Vicsek. 1994. "Generic modelling of cooperative growth patterns in bacterial colonies". *Nature,* vol. 368, no. 6466, 46-49.

Elfwing, A.; Y. LeMarc; J. Baranyi; and A. Ballagi. 2004. "Observing growth and division of large numbers of individual bacteria by image analysis". *Applied and Environmental Microbiology,* vol. 70, no. 2, 675-678.

Ginovart, M.; D. Lopez; and J. Valls. 2002. "INDISIM, an individual-based discrete simulation model to study bacterial cultures". *Journal of theoretical biology,* vol. 214, no. 2, 305-319.

Ginovart, M.; D. Lopez; J. Valls; and M. Silbert. 2002. "Individual based simulations of bacterial growth on agar plates". *Physica A: Statistical Mechanics and its Applications,* vol. 305, no. 3-4, 604-618.

Jennings, N.R.; K. Sycara; and M. Wooldridge. 1998. "A Roadmap of Agent Research and Development". *Autonomous Agents and Multi-Agent Systems,* vol. 1, no. 1, 7-38.

Kreft, J.U.; G. Booth; and J.W. Wimpenny. 1998. "BacSim, a simulator for individual-based modelling of bacterial colony growth". *Microbiology (Reading, England),* vol. 144 (Pt 12), no. Pt 12, 3275-3287.

Lacasta, A.M.; I.R. Cantalapiedra; C.E. Auguet; A. Penaranda; and L. Ramirez-Piscina. 1999. "Modeling of spatiotemporal patterns in bacterial colonies". *Physical Review. E, Statistical Physics, Plasmas, Fluids, and Related Interdisciplinary Topics,* vol. 59, no. 6, 7036-7041.

Matsumoto, M. and T. Nishimura. 1998. "Mersenne twister: a 623-dimensionally equidistributed uniform pseudo-random number generator". *ACM Trans. Model. Comput. Simul.*, vol. 8, no. 1, 3-30.

Mitchell, J.G. and K. Kogure. 2006. "Bacterial motility: links to the environment and a driving force for microbial physics". *FEMS microbiology ecology,* vol. 55, no. 1, 3-16.

Niven, G.W.; T. Fuks; J.S. Morton; S.A. Rua; and B.M. Mackey. 2006. "A novel method for measuring lag times in division of individual bacterial cells using image analysis". *Journal of microbiological methods,* vol. 65, no. 2, 311-317.

Nystrom, T. 2001. "Not quite dead enough: on bacterial life, culturability, senescence, and death". *Archives of Microbiology,* vol. 176, no. 3, 159-164.

Ross, T.; D.A. Ratkowsky; L.A. Mellefont; and T.A. McMeekin. 2003. "Modelling the effects of temperature, water activity, pH and lactic acid concentration on the growth rate of Escherichia coli". *International journal of food microbiology,* vol. 82, no. 1, 33-43.

Segall, J.E.; S.M. Block; and H.C. Berg. 1986. "Temporal comparisons in bacterial chemotaxis". *Proceedings of the National Academy of Sciences of the United States of America,* vol. 83, no. 23, 8987-8991.

Siegele, D.A. and R. Kolter. 1992. "Life after log". *Journal of Bacteriology,* vol. 174, no. 2, 345-348.

Swinnen, I.A.; K. Bernaerts; E.J. Dens; A.H. Geeraerd; and J.F. Van Impe. 2004. "Predictive modelling of the microbial lag phase: a review". *International journal of food microbiology,* vol. 94, no. 2, 137-159.

Walshe, R. 2006. "Modelling bacterial growth patterns in the presence of antibiotic". In *Proceedings of the 11th IEEE International Conference on Engineering of Complex Computer Systems* (Stanford University, CA, Aug.15-17). IEEE Computer Society, Washington, DC, 177-186.

WATER ANOXIA AND SPECIES SELECTION IN LAGOONS: AN ANALYSIS OF ECOSYSTEM DYNAMICS

Francesco Cioffi and Giovanni Cannata
Dipartimento di Idraulica, Trasporti e Strade
Università degli Studi di Roma 'La Sapienza'
Via Eudossiana 18, 00184, Rome, Italy
E-mail: Francesco.cioffi@uniroma1.it

KEYWORDS
Lagoons, Mathematical models, eutrophication, ecosystem dynamics.

ABSTRACT

In order to understand the complicate cause-effect link between eutrophication trend on the long run, type of dominant vegetal species, and instantaneous conditions leading to summer water anoxia, the dynamic behaviour of a lagoon ecosystem is investigated by using an eutrophication model. Phosphorous external load is assumed as control parameter. The response diagram, obtained by simulation, varying the value of control parameter, shows the existence of different ranges of stability of the ecosystem, characterized by the dominance of a specific group of primary producers and by a different ecosystem vulnerability to summer water anoxia. A catastrophic bifurcation occurs for a critical value of the control parameter which manifests as an abrupt change of the dominant species from eelgrass to macroalgae. The serious consequences of such a selection in terms of eutrophication processes and anoxic crisis vulnerability of lagoons are emphasized.

INTRODUCTION

One of the most evident effects of eutrophication in coastal lagoons is the change, in the long run, in vegetable species with reduction of diversity and prevalence of infesting species; such change is also accompanied by an increase of the summer water anoxia vulnerability of the lagoons.

Recent investigations have shown that a correlation exists between the lagoon eutrophication levels, vulnerability to water anoxia and type of dominant vegetable species: coastal lagoons having lower eutrophication levels, in which Summer water anoxia events are rare or absent, are mainly colonised by rooted species (as eelgrass), while in highly eutrophic environments, in which more frequently water anoxia phenomena occur, different floating species, from macroalgae to microalgae, are dominant.

Lagoon ecosystems exhibit a complex dynamic which is due to non linear interactions of biological , chemical and hydrodynamic processes that influence the cycles of carbon (vegetal growth, organic detritus production and mineralization) , nutrients , sulphur, and of all those species playing any role in ecological phenomena (Back 1993), (Coffaro et.al. 1997),(Ravel et.al. 2003),(Sfriso et.al. 2005). Such a dynamic is influenced by various external forcing variables - tidal flow rate, wind speed, temperature, light intensity, nutrient external load - that in most of the cases are periodical or multi-periodical. Anthropogenic or natural changes in one o more of these external variables can trigger shifts between ecosystem states which can be evidenced by species composition shift, higher vulnerability of the lagoon to summer water anoxia, more frequent fish kills. These regime shifts can be smooth, abrupt or discontinuous; furthermore they may not be immediately reversible (Carpenter et.al. 1999), (Collie et.al. 2004).

The understanding of the mechanisms driving the regime shifts, i.e. of the cause-effect link between eutrophication trend on the long run, type of dominant vegetal species, and instantaneous conditions leading to summer water anoxia, appears to be fundamental to individuate strategies to management lagoon ecosystems.

In this paper the ecosystem dynamic in lagoons, as a consequence of nutrient enrichment, was investigated by an eutrophication model (Cioffi and Gallerano 2006), (Cioffi and Gallerano 2001). The rate of phosphorous external load was assumed as control parameter. A diagram representing the ecosystem dynamic response to the changes of the control parameter was constructed by simulations. Such a response diagram suggests the existence of regime shifts which border different ecosystem stability ranges, characterized by the dominance of a specific group of primary producers and by different ecosystem vulnerability at summer water anoxia. A catastrophic bifurcation or abrupt regime shift - evidenced by Poincarè sections obtained mapping, at a prefixed time period, the values of the concentration of the species - occurs for a critical value of the control parameter and manifests itself as an abrupt change of the dominant species from eelgrass to macroalgae.

HYPOTHESIS AND MODEL EQUATIONS

The model simulates, in water column and in sediments, the concentration temporal evolution of the species: dissolved oxygen, vegetal organic carbon, particulate and dissolved organic carbon, orthophosphate, hydrogen sulphide, adsorbed phosphorous (Cioffi and Gallerano 2006), (Cioffi and Gallerano 2001). Such a model is applied to an hypothetical and schematic lagoon, having superficial area A, tidal flow rate q, and a constant depth h. Horizontal homogeneity is assumed and the two main hydrodynamic effects on eutrophication processes - turbulent diffusion and flushing of the species from the lagoon toward the sea- are represented respectively by vertical turbulent diffusion terms,

(whose turbulent diffusion coefficient is related to the instantaneous value of tidal flow rate and wind speed (Cioffi and Gallerano 2006)) - and sink/source terms related to the tidal flow rate and to the volume of the lagoon.
The mass balance equations of the model are the following:

$$\begin{aligned}\frac{\partial C_{al}}{\partial t} &= \frac{\partial}{\partial z}\left(\nu_t \frac{\partial C_{al}}{\partial z}\right) \\ &+\begin{pmatrix}\mu_{cral}\cdot f_a(P_o)\cdot f(I)\cdot f_{al}(T)-r_{al}f_{al}(T) \\ -\frac{K_{d\,al}}{f_{al}(T)}-\frac{K_{p\,al}}{f_{al}(T)}\end{pmatrix}\cdot C_{al} \\ &+\alpha_{al}\cdot(q/(A\cdot h))\cdot sin(\omega_m t)\cdot\tilde{C}_{al}\end{aligned} \tag{1}$$

$$\begin{aligned}\frac{\partial C_{ph}}{\partial t} &= \frac{\partial}{\partial z}\left(\nu_t \frac{\partial C_{ph}}{\partial z}\right) \\ &+\left(\mu_{cph}\cdot f_{ph}(P_o)\cdot f(I)\cdot f_{ph}(T)\right. \\ &\left.-r_{ph}f_{ph}(T)-\frac{K_{d\,ph}}{f_{ph}(T)}-\frac{K_{p\,ph}}{f_{ph}(T)}\right)\cdot C_{ph} \\ &+(q/(A\cdot h))\cdot sin(\omega_m t)\cdot\tilde{C}_{ph}\end{aligned} \tag{2}$$

$$\begin{aligned}\frac{d\tilde{C}_m}{dt} &= \int_o^h\left(\mu_{crm}\cdot f_s(P_o,P_{ads})\cdot f(I)\cdot f_m(T)\right. \\ &\left.-r_m\cdot f_m(T)-\frac{K_{d\,m}}{f_m(T)}-\frac{K_{p\,m}}{f_m(T)}\right)\cdot C_m dz\end{aligned} \tag{3}$$

$\tilde{C}_m = \int_0^h C_m(z)dz$ with $C_m(z) < C_{max}$

$$\begin{aligned}\frac{\partial C_d}{\partial t} &= \frac{\partial}{\partial z}\left(\nu_t \frac{\partial C_d}{\partial z}\right)+\frac{K_{d\,al}}{f_{al}(T)}\cdot C_{al}+\frac{K_{d\,m}}{f_m(T)}\cdot C_m \\ &+\frac{K_{d\,ph}}{f_{ph}(T)}\cdot C_{ph}-\mu_d\cdot f_\mu(T)\cdot f(C_d)\cdot f(O_2) \\ &+K_p C_p\alpha_{por}-K_s C_d+(q/(A\cdot h))\cdot sin(\omega_m t)\cdot\tilde{C}_d \\ &+q_{dmm}\end{aligned} \tag{4}$$

$$\begin{aligned}\frac{\partial C_p}{\partial t}-\nu_s\frac{\partial C_p}{\partial z} &= \frac{\partial}{\partial z}\left(\nu_t \frac{\partial C_p}{\partial z}\right)-K_p C_p \\ &+\frac{K_{p\,al}}{f_{al}(T)}\cdot C_{al}+\frac{K_{p\,m}}{f_m(T)}\cdot C_m+\frac{K_{p\,ph}}{f_{ph}(T)}\cdot C_{ph} \\ &+(q/(A\cdot h))\cdot sin(\omega_m t)\cdot\tilde{C}_p+q_{pimm}\end{aligned} \tag{5}$$

$$\begin{aligned}\frac{\partial P_O}{\partial t} &= \frac{\partial}{\partial z}\left(\nu_t \frac{\partial P_O}{\partial z}\right) \\ &+K_{p_c}\left(\mu_d f_\mu(T)f(C_d)f(O_2)+K_s C_d\right) \\ &-K_{pc}\left(\mu_{cral}\,f_a(P_o)f(I)f_{al}(T)-r_{al}\,f_{al}(T)\right)\cdot C_{al} \\ &-K_{pc}\left(\mu_{cph}\cdot f_{ph}(P_o)\cdot f(I)\cdot f_{ph}(T)\right. \\ &\left.-r_{ph}f_{ph}(T)\right)\cdot C_{ph}-\frac{1-p_{or}}{p_{or}}\cdot K_a(P_{ae}-P_a) \\ &+\alpha_p\left(\frac{1-p_{or}}{p_{or}}\right)-\left[\mu_{crm}\cdot f_s(P_o,P_{ads})\cdot f(I)\cdot f_m(T)\right. \\ &\left.-f_m(T)\cdot r_m\right]\cdot\frac{\tilde{C}_m K_{pmc}}{h_s}\cdot f'(z) \\ &+(q/(A\cdot h))\cdot sin(\omega_m t)\cdot\tilde{P}_o+q_{qp_o imm}\end{aligned} \tag{6}$$

$$\begin{aligned}\frac{dP_a}{dt} &= K_a(P_{ae}-P_a)-\alpha_p-\left(\mu_{crm}\cdot f_s(P_o,P_a)\cdot f_m(I)\cdot f_m(T)\right. \\ &\left.-f_m(T)\cdot r_m\right)\cdot\frac{\tilde{C}_m K_{pmc}}{h_s}\cdot f''(z)\end{aligned} \tag{7}$$

$$\begin{aligned}\frac{\partial H}{\partial t} &= \frac{\partial}{\partial z}\left(\nu_t \frac{\partial H}{\partial z}\right)-K_H OH+ \\ &+\alpha_s\cdot K_s C_d+(q/(A\cdot h))\cdot sin(\omega_m t)\cdot\tilde{H}\end{aligned} \tag{8}$$

$$\begin{aligned}\frac{\partial O}{\partial t} &= \frac{\partial}{\partial z}\left(\nu_T \frac{\partial O}{\partial z}\right)+\alpha_\mu\cdot\left(\mu_{cph}\cdot f_{ph}(P_o)\cdot f(I)\cdot f_{ph}(T)\right. \\ &\left.-r_{ph}f_{ph}(T)-\frac{K_{d\,ph}}{f_{ph}(T)}-\frac{K_{p\,ph}}{f_{ph}(T)}\right)\cdot C_{ph} \\ &+\alpha_\mu\cdot\left(\mu_{cral}\cdot f_a(P_o)\cdot f(I)\cdot f_{al}(T)-r_{al}\cdot f_{al}(T)\right. \\ &\left.-\frac{K_{d\,al}}{f_{al}(T)}-\frac{K_{p\,al}}{f_{al}(T)}\right)\cdot C_{al} \\ &+\alpha_\mu\cdot\left(\mu_{crm}\cdot f_s(P_o,P_a)\cdot f(I)\cdot f_m(T)\right. \\ &\left.-r_m\cdot f_m(T)-\frac{K_{d\,m}}{f_m(T)}-\frac{K_{p\,m}}{f_m(T)}\right)\cdot C_m \\ &-\beta_1\cdot\mu_d f_\mu(T)\cdot f(C_d)\cdot f(O_2)-\beta_2 K_H OH \\ &+\alpha_o(q/(A\cdot h))\cdot sin(\omega t)\cdot\tilde{O}\end{aligned} \tag{9}$$

The meaning of the symbols of Eqs (1-9) is reported in appendix. Eqs (1-3) refer to the three main groups of primary producers (Sfriso et.al. 2005) - eelgrass C_m (as Zostera), macro-algae C_{al} (as Ulva) and micro-algae or phytoplankton C_{ph} (as Chlorella) - whose vegetal growth, by photosynthesis, is controlled by environmental factor – light, temperature, lagoon hydrodynamics - and by the physiological characteristics of the particular vegetal species; the three species differ respect to nutrient uptake, growth and organic detritus production rates. The eelgrass (as Zostera) is a rooted vegetal species, while the other two are floating ones; eelgrass assimilates phosphorous mainly by root-rhizomes from the interstitial water of the sediment (Back 1993) and it growths starting from the bottom of the water column, then the vegetation extends to the upper layers of the water column once a maximum vegetal concentration (C_{max} in Eq. (3)) is reached in the lower layers. Floating species uptake phosphorous in the water column and they growth mainly near the free surface of the water column where there are more favourable light conditions. In Eqs(1 -3) the nutrient uptake is represented by Michaelis-Menten kinetics, $V=V_{max}\ (P/(P+K_{s)})$, where V is the uptake velocity, V_{max} the maximum uptake velocity (which directly related to the algae growth rate), P the concentration of nutrient and K_s the half-saturation constant. Low values of K_s are thought to express a competitive advantage for algal species at low nutrient concentrations, whilst those algae with higher values of V_{max} are considered to be favoured at higher concentrations. Experimental studies in aquatic ecosystems (Wen et.al. 1997) have shown that both the values of V_{max} and K_s depend on the algae size (or by surface/volume ratio, S/V): the maximum PO_4 uptake small algae capacity per unit volume is higher than large algae one, whereas K_s decreases with increasing algae size. The rate of production of organic detritus by rooted plants is lower than in floating species and it has appreciable values only at the end of the eelgrass life cycle; on the contrary the floating species, having a quickly turn over, during the entire life cycle, show a greater and more uniform detritus production rates. These rates increase with increasing of algae S/V ratio. On the basis of these observations the vegetal growth rates, the organic detritus production rates and the half-saturation constant were related to the S/V ratio (Coffaro et.al. 1997). Eqs(4-5) represent the mass balances, in water and sediments, of both dissolved C_d and particulate organic detritus C_p ; in Eqs (4-5) different processes are represented: vegetal organic detritus production, transformation of C_p in C_d , settling, aerobic and anaerobic mineralization. Eqs (6-7) represent the dissolved phosphorous (P_o) mass balance, in water and in sediments, and the adsorbed phosphorous (P_a) mass balance in sediments. The following processes are represented: aerobic and anaerobic adsorbing-desorbing of dissolved phosphorous in sediments, phosphorous uptakes by vegetal species, source of phosphorous by aerobic and anaerobic mineralization. Eq (8) represents the mass balance of sulphide (H) produced by anaerobic mineralization processes (sulphide reduction) and consumed by re-oxidation. Eq (9) represents the dissolved oxygen (O) mass balance as a consequence of photosynthesis, algal respiration, aerobic mineralization, sulphide re-oxidation.

Periodical forcing terms appear into the model equations. They represent the influences of external variables - as light, temperature, tidal flow rate, breeze speed- having some typical periodicity or multi-periodicity: light and temperature have both seasonal and daily variation, tidal flow rate has a semi-daily variation, and breeze speed has a daily variation.

Semi-discretization of the diffusive terms in Eqs (1-9) in the space variable z, on discrete points along the vertical, by using a centered finite difference scheme, yields a nN ODE system, being n the number of points of the grid and N the species number. A non equidistant grid, obtained by applying a cosine function and having 11 points in the water column and 6 into the sediment layer, was used in order to obtain a better approximation of the derivates close to the interface air-water and water-sediment. The ODE system, for assigned boundary conditions (see (Cioffi and Gallerano 2006),(Cioffi and Gallerano 2001)), was solved using a forth order Runge-Kutta method. In order to implement such a method, a fortran code was constructed. A time step of 100 s was used in the simulations.

PROCEDURE AND RESULTS

In order to analyze the response of the ecosystems to the variations of nutrient load discharged in the lagoon, and to investigate on the relationship between summer water anoxia phenomena and eutrophication state of the lagoon, the following procedure was carried out:

a) First a set of biological, chemical and physical parameters (on the basis of literature) and external forcing variables was selected; the parameter values, which define the parameter space, are reported in appendix;
b) the phosphorous load q_{poimm} (gr/m^3s) was assumed as control parameter whose value has been changed within a prefixed range;
c) for each selected value of the control parameter a simulation by model was carried out as long as periodic solutions of the equation system were reached; the existence of periodical solutions was verified observing the evolution of the system in the phase space projected to significant couples of variables of the system. In the simulations a critical summer condition having a four days period of absence of breeze was assumed;
d) a Poincarè section was constructed mapping the values of the concentration of the species, describing the state of the system, at a prefixed time for each simulated year; this time was chosen at the end of July, at the 6.00 a.m, corresponding to the critical summer condition. It should be noted that a periodic solution of the dynamic systems , which in the phase space is described by a limit cycle, in a Poincarè section appears as a fixed point;
e) a response (or bifurcation) diagram, depicting the values at the fixed points of the more significant variables vs. the control parameter values was constructed.

In fig. 1. the response diagram is shown ; in the diagram the values, corresponding to the fixed points, of three vegetal species concentrations, integrated along the water column depth , and of the dissolved oxygen concentration at the bottom of the water column, are reported vs. phosphorous load (gr/m^3 s).

The figure shows that, varying the phosphorous load, different lagoon trophic conditions occur. These conditions are characterized by the dominance of specific vegetal species. In fact, in figure 1, it is possible to identify three regions defined by the values of the control parameter q_{poimm}:

a) $q_{poimm} < 10^{-8}$ gr/m^3s, region I: eelgrass is the only species present in lagoon; values of dissolved oxygen concentration are generally greater than 4 mg/l ; no Summer water anoxia occurs;

b) $10^{-8} < q_{poimm} < 10^{-7}$ gr/m^{3}s, region II: macroalgae is the only dominant species; for values of $q_{pimm} > 2\ 10^{-8}$ the values of minimum Summer dissolved oxygen concentration are practically null; this means that lagoon is subject each year to water anoxia phenomena;

c) $10^{-7} < q_{poimm}$ gr/m^{3}s, region III: macroalgae and microalgae coexists. For $q_{poimm} > 8\ 10^{-7}$ microalgae is the dominant species. Also in this region the minimum summer dissolved oxygen concentration values are null denoting water anoxia phenomena.

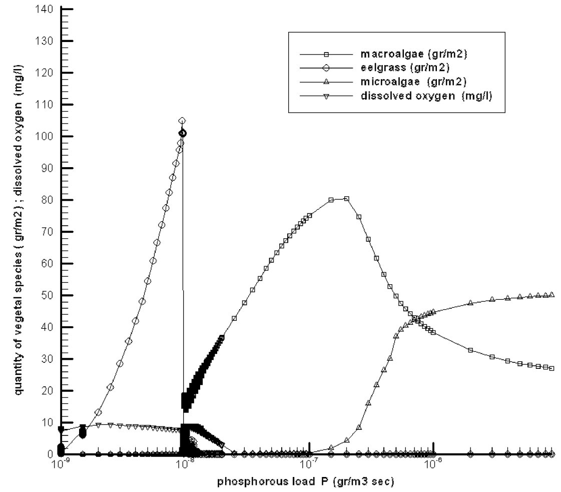


Figure 1 Maximum values of vegetal species concentration vs. phosphorous load

The evolution in time of the ecosystems for an assigned value of the control parameter can be analyzed in the phase space projected to two or more significant species. The adsorbed phosphorous concentration vs. the eelgrass concentration is shown in the phase plane in figure 2.

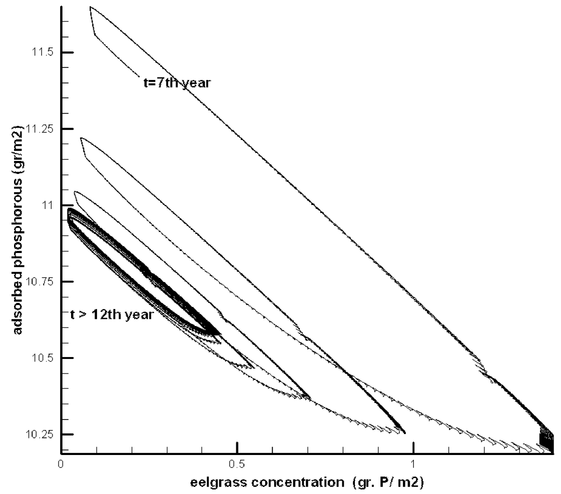


Figure 2 Phase plane: adsorbed phosphorous vs. eelgrass concentration (q_{poimm}=5 10^{-9} gr/m^3)

Figure 2 is obtained by the calculated temporal series of the species concentrations. An example of these temporal series is shown in figure 3. Figures 2 and 3 refers to the simulation with q_{poimm}=5 10^{-9}. Figure 2 represents a typical behaviour of the ecosystem in the region I: after the transitory, a periodical solution is obtained, clearly evidenced by the existence of the limit cycle shown in figure 2.

In figure 2 the higher values of the concentration of eelgrass correspond to lower values of adsorbed phosphorous concentration in sediments and vice versa: in fact eelgrass uptakes phosphorous from sediments, thus the quantity of adsorbed phosphorous reduces; at the end of the eelgrass life cycle the dissolved phosphorous produced by detritus mineralization is ready transformed in adsorbed phosphorus; the good oxygenation conditions of the environmental favourite the last process. These good oxygenation conditions can be evidenced by the phase plane of figure 4 where the dissolved oxygen concentrations at the bottom of the water vs. eelgrass concentrations, for simulation time > 12 years, are shown.

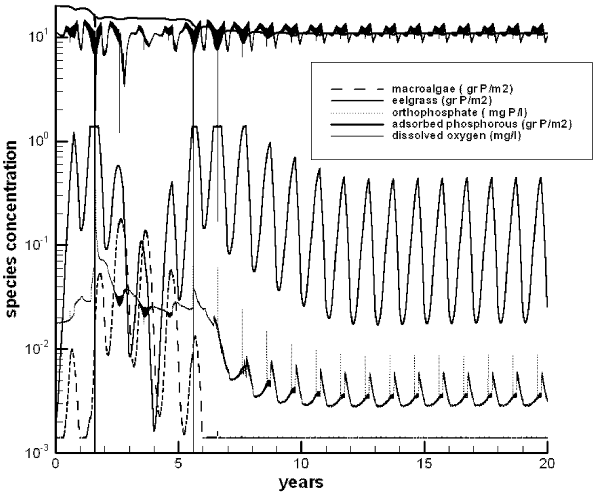


Figure 3. Significant species concentration vs. time (q_{poimm}=5 10^{-9} gr/m^3)

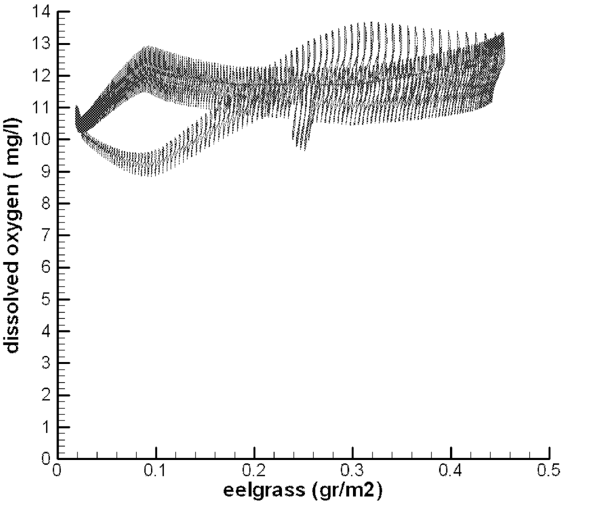


Figure 4. Phase plane: dissolved oxygen concentration vs. eelgrass for t> 12 years (q_{poimm}=5 10^{-9} gr/m^3)

It should be noted (see figure 3) that in the first region the orthophosphate concentration in water column is so low to prevent the floating algae growth.

The phase plane in figure 5 shows the orthophosphate concentration in water column vs. macroalgae concentration; figure 5 refers to a simulation in which a phosphorous load value within the region II was assumed (q_{poimm}=5 10^{-8}); from the figure 5 it is possible to observe that, after the transitory, the trajectories collapse in a limit cycle. From the figure it is possible to observe that, in coherence with the macroalgae growth cycle, at higher macroalgae concentration values correspond lower orthophosphate concentration values and vice versa. In region II macroalgae is the dominant vegetal species.

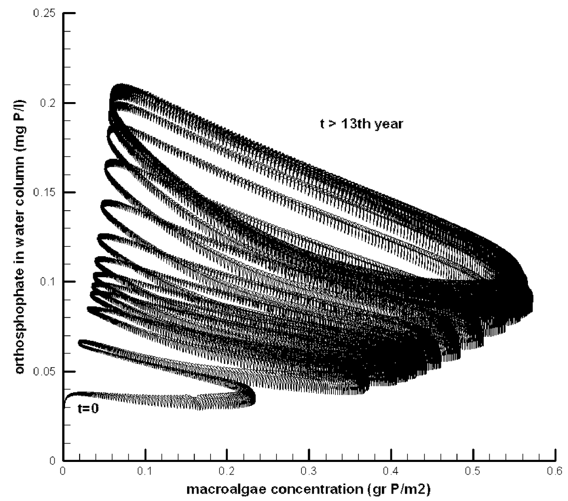


Figure 5 Phase plane orthophosphate vs. macroalgae (q_{poimm}=5 10^{-8} gr/m^3)

This can be explained observing that, in region II, (see figure 6) the orthophosphate concentrations in the water column, resulting from the balance between the phosphorous discharged into the lagoon and the one flushing from the lagoon to the sea, are high enough to allow the macroalgae growth, but not so high to support the microalgae growth. Furthermore eelgrass growth is inhibited by the shadowing due to the presence of macroalgae (Cioffi and Gallerano 2006).

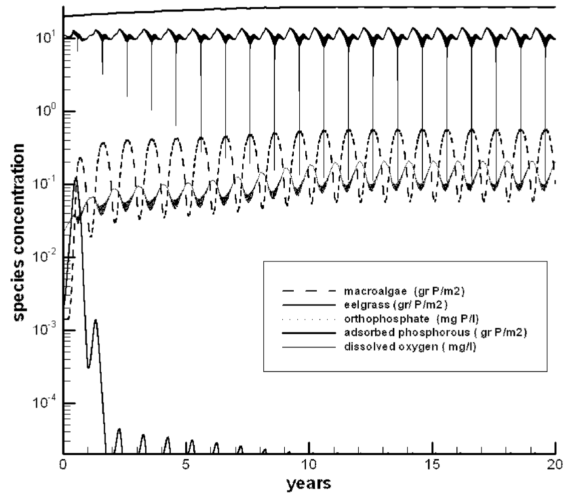


Figure 6 Significant species concentration vs. time (q_{poimm}=5 10^{-8} gr/m^3)

From figure 6 it can be observed that ,starting from the 10th year, during the critical summer period when the wind speed falls down, the dissolved oxygen concentration at the water column bottom drops close to zero, i.e. summer water anoxia occurs. This phenomenon is also evidenced in the limit cycle shown in figure 7.

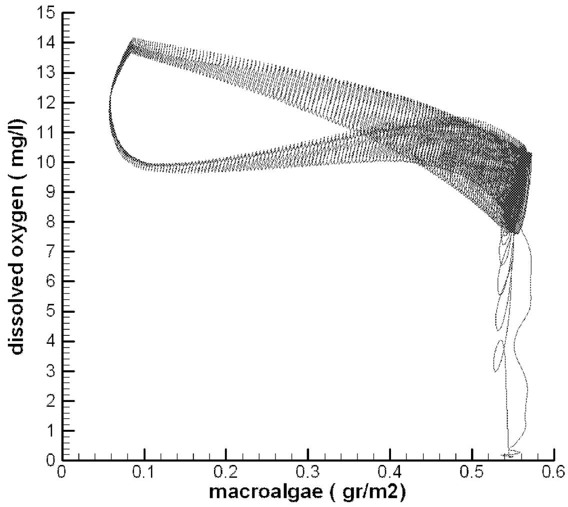


Figure 7 Phase plane dissolved oxygen vs. macroalgae for t> 12 years (q_{poimm}=5 10^{-8} gr/m^3)

Comparing figure 6 with figure 3 , or figure 4 with figure 7, it is possible to observe that ,even if the maximum quantities of vegetal organic carbon for eelgrass and macroalgae are about the same, the summer water anoxia occurs only in the case in which macroalgae are the dominant species. This result seems to demonstrate that a relationship exists between kind of vegetal species and summer water anoxia in lagoons. A possible explanation can be found observing that the production of organic detritus is different in eelgrass and in macroalgae. In the case of rooted plants, as eelgrass, the organic detritus is produced at the end of the life cycle, in Autumn, when leaves fall down; while floating species have a more rapid turn-over and therefore the organic production is more uniform during the life cycle (from the end of winter to Autumn). Therefore, even if the floating macroalgae have about the same concentration of the eelgrass, the cycles of organic matter, phosphorous and sulphur are completely altered. More organic detritus is produced by floating algae, and more organic detritus settles and accumulates into sediments from the end of winter to the critical summer season. This determines a major oxygen requirement for the mineralization of organic matter and therefore more reducing conditions into sediments, which determinates a greater vulnerability of lagoon to summer water anoxia.

In figure 8 the two limit cycles are shown related to, respectively, macroalgae and microalgae concentration vs. orthophosphate concentration in water column.

Figure 8 refers to a phosphorous load value (q_{poimm}=5 10^{-7}) within the region III. The two limit cycles present a very complex feature and no clear relation exists between the concentration of available phosphorous for algae growth and algae concentration , like in the cases of figure 2 and 5. This is due to the fact that ,during the entire Summer period, frequent water anoxia occur also in presence of breeze winds; such a quasi- permanent water anoxia condition determinates the release from sediment to water column of the adsorbed phosphorous in dissolved phosphorous. The phosphorous, released from sediments, affects the phosphorous balance in the water column masking the effect of algae uptakes.

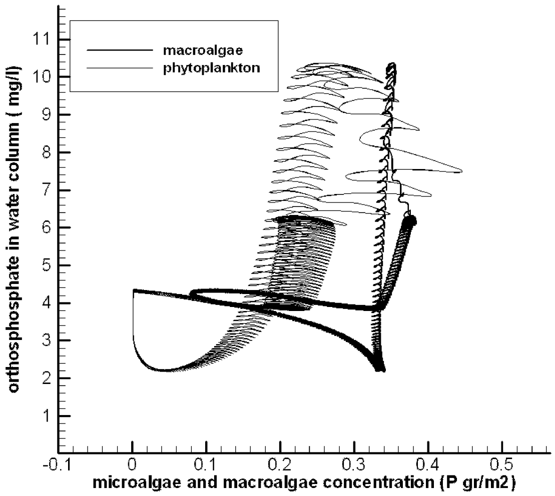


Figure 8 Phase plane orthophosphate vs. microalgae and/or macroalgae for t> 15 years (q_{poimm}=5 10^{-7} gr/m^3)

Figure 1 shows that an abrupt change in vegetal species composition, from eelgrass to macroalgae, occurs within the small range of values 9 10^{-9} <q_{poimm}< 10^{-8} gr/m^{3}s. In order to investigate this qualitative and 'catastrophic' change in the ecosystem behaviour numerous simulations were carried out, in the previously indicated range, varying the control parameter of a quantity Δq_{poimm}=2 10^{-11} gr/m^{3}s. Thus Poincarè sections were constructed mapping, at prefixed times with period T= 1 year, the values of the concentration of the species (see the description of point d)). In figures 9 and 10 the most significant results of this analysis are shown. Figure 9 represents the Poincarè section related to eelgrass vs. macroalgae concentrations.

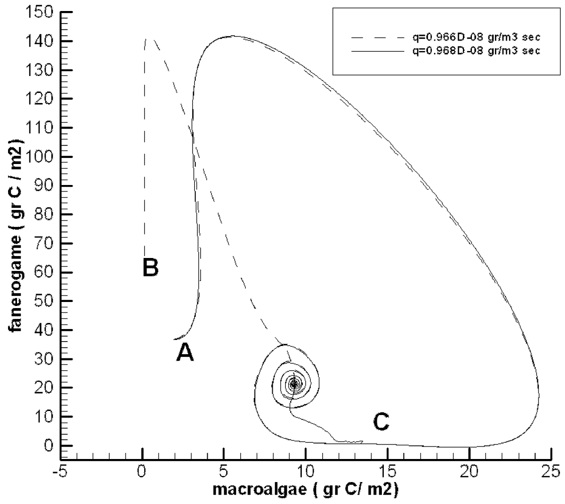


Figure 9 Poincarè section: eelgrass vs. macroalgae

The letter A in figure 9 indicates the point in which the trajectories describing the evolution of the state of the ecosystem in the first year of simulation intersects the Poincarè section; the letter B and C in figure 9 represent the fixed points, corresponding to limit cycles in the phase space, obtained by simulations in which two very close values of q_{poimm} were assumed (q_{poimm}=9.66 10^{-9} and q_{poimm}=9,68 10^{-9}). The two curves in figure 9 describe very close trajectories which form a spiral cycle , but once in the center of the spiral the two curves move in opposite directions towards the respective fixed points.

This bifurcation behaviour is a direct consequence of the nonlinearity of the equation system representing the ecosystem. It should be deducted by figure 9 that eelgrass and macroalgae are mutually exclusive; in fact the fixed point B represents a state of the ecosystem in which only eelgrass is present, whereas, the fixed point C represent a state of the ecosystem in which macroalgae has completely replaced the eelgrass.

In order to complete the analysis, in figure 10 the Poincarè section, constructed with the values of orthophosphate concentration vs. adsorbed phosphorous concentration, is shown; the figure 10 refers to the same simulations of figure 9. The two fixed points B and C in figure 10 refer to the same state of the ecosystems of the fixed points B and C in figure 9. Point B in figure 10 represents the state in which eelgrass is present, the dissolved phosphorous concentration is low, and the adsorbed phosphorous has been up-taken by eelgrass; while point C in figure 10 represents a state of the system in which the concentration of dissolved phosphorous is high enough to support the macroalgae growth; further the adsorbed phosphorous concentration is in chemical equilibrium with the dissolved one.

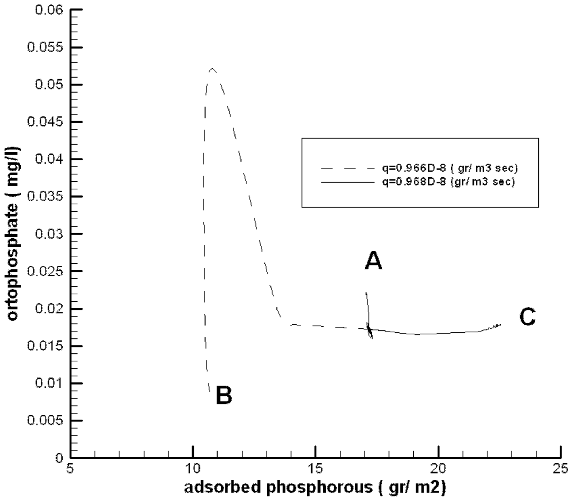


Figure 10 Poincarè section: orthophosphate vs. adsorbed phosphorous

CONCLUSION

The analysis of the dynamic behaviour of a lagoon ecosystem conducted by the eutrophication model has evidenced: a) the existence of different ranges of stability of the ecosystem depending on the value of the external phosphorous load and characterized by the dominance of a specific group of primary producers; b) in the ranges of values of the external phosphorous load where floating algae are dominant the ecosystem has a greater vulnerability to Summer water anoxia. This can be justified on the bases of the different production modalities and rates of organic detritus of floating algae, micro and macroalgae, with respect to eelgrass; c) a catastrophic bifurcation occurs for a critical value of the control parameter which manifests with an abrupt substitution of eelgrass with macroalgae.

Even if the results are encouraging the analysis conducted by simulations allows individuating only the stable branches of response diagram; it does not allow the individuation of unstable branches; further it does not allow the geometrical characterization of bifurcation points. This characterization can be performed using continuation techniques. An attempt to apply these techniques to lagoon ecosystems is in progress.

REFERENCES

Back H.K. 1993. "A dynamic model describing the seasonal variations in growth and the distribution of eelgrass (Zostera marina L.)." *Ecological modelling*, **65**, 31-50.

Carpenter S.R., Ludwing D., Brock W.A. 1999. "Management of eutrophication for lake subject to potentially irreversible change." *Ecological Application*, **9(3)**, pp.751-771.

Cioffi F., Gallerano F. 2006. "From rooted to floating vegetal species in lagoons as a consequence of the increases of external nutrient load: An analysis by model of the species selection mechanism." *Applied Mathematical modelling*, **30**, 10-37.

Cioffi F., Gallerano F. 2001. "Management strategies for the control of eutrophication processes in Fogliano lagoon (Italy) a long term analysis using a mathematical model." *Applied Mathematical modelling*, **25**, 385-426.

Coffaro G., Bocci M., Bendoricchio G. 1997. "Application of structural dynamic approach to estimate space variabilità of primary producers in shallow marine water." *Ecological Modelling*, **102**, 97-114.

Collie J.S., Richardoson K., Steele J.H. 2004. "Regime shift: can ecological theory illuminate the mechanism?" *Progress in Oceanography*, **60,** 281-302.

Raven J.A., Taylor R. 2003. "Macroalgae growth in nutrient-enriched estuaries: A biogeochemical and evolutionary perspective." *Water, air and soil pollution*, **3**: 7-26.

Sfriso A., Facca C., Ceoldo S.,Marcomini A. 2005. "Recording the occurrence of trophic level changes in the lagoon of Venice over '90s." *Environment international*, **31**, 993-1001.

Wen Y.H., Vezina A., Peters R.H. 1997. "Allometric scaling of compartmental fluxes of phosphorous in freshwater algae." *Limnol. Oceanogr.*, **42**, 45-56.

APPENDIX

External forcing factor: A: lagoon surface (km^2) 3.0; q: tidal flow rate (m^3/s) 4; breeze speed (m/s) 3.5

C_{max} : maximum eelgrass concentration (gr C/l)=100

D_{mo} , D_{mn} , O2, H2S diffusion coefficients (m^2/s) 10^{-10}

$\widetilde{C}_{al}, \widetilde{C}_{ph}, \widetilde{C}_d, \widetilde{C}_p, \widetilde{P}_o, \widetilde{H}, \widetilde{O}$: species concentration at the lagoon outlet, equal to zero for entering tidal flow rates and equal to: $C_{al}, C_{ph}, C_d, C_p, P_o, H, O$, for going out tidal flow rates.

$f_{al}(P_o) = P_o / (K_{poal} + P_o)$,

$f_{ph}(P_o) = P_o / (K_{poph} + P_o)$,

$f_s(P_o, P_a) = (P_o + P_a) / (K_{pos} + P_o + P_a)$

nutrient limiting factors;

$f(O) = O / (K_o + O)$ oxygen limiting factor;

$f(C_d) = C_d / (K_d + C_d)$ dissolved organic carbon limiting factor;

$f(I) =$

$(I / I_m) \cdot sin(\pi \cdot \omega_y \cdot (t - t_s)) sin(\pi \cdot \omega_d \cdot (t - t_s)) \cdot e^{-\gamma \cdot (h - z)}$

light extinction;

$f_{al,ph}(T) = K_{tal,ph}{}^{(T - T_{al,ph})}$,

$f_m(T) = (T / T_m) \cdot e^{(1 - (T / T_m))}$,

$f_\mu(T) = K_{t\mu}{}^{(T - T_\mu)}$ temperature limiting factors;

h: water column depth (m) 1.20;

h_s: sediment layer depth (m.) 0.1;

K_a; aerobic adsorbing-desorbing phosphorus rate in sediment (s^{-1}) 2.5 10^{-6};

K_d: half saturation constant (aerobic mineralization) (mg/l) 10;

K_{dal}, K_{dph}, K_{dm}: dissolved organic carbon production rate (s^{-1}) 10^{-7}, 10^{-6},5 10^{-8} ;

K_H : reoxidation rate of Hydrogen sulphide (s^{-1}) 4.2 10^{-6};

K_o: half saturation constant limiting aerobic mineralization (mg/l) 0.3;

K_{pc} , K_{pmc} (mg/l PO_4^-)/(mg/l$C_{al, ph, d}$) 0.01

K_p particulate organic in dissolved organic carbon rate (s^{-1}) 5 10^{-7};

K_{pal},K_{pph}, K_{pm} particulate organic carbon production rate (s^1) 10^{-7}, 10^{-6}, 5 10^{-8};

K_{po} adsorbed-dissolved phosphorus half saturation constant (mg/l) 510^{-3};

K_{poal}, K_{poph}, K_{pos} half saturation constants (mgPO_4/l) 0.02, 0.08, 0.01;

K_s organic carbon mineralization rate in anaerobic conditions (s^{-1}) 2.5 10^{-7};

$K_{\gamma 1}$; $K_{\gamma 2}$; $K_{\gamma 3}$: $K_{\gamma 4}$ extinction light coefficients 0.06, 0.05, 0.005, 0.06;

$P_{ae}= P_{max} (P_o /(P_o + K_{po}))$ equilibrium adsorbed phosphorous concentration;

P_{max}: maximum adsorbed phosphorus concentration in sediments (mg/l) 700;

p_{or} : porosity = 0,8;

q_{pimm}, q_{dmm}, q_{poimm} : external loads of particulate and dissolved carbon and phosphorous;

r_{al} , r_{ph} , r_m :vegetal respiration rate (s^{-1}) 6 10^{-7}, 5 10^{-6}, 3 10^{7}

α_{al} hydrodynamic factor 0.05;

α_p : anaerobic adsorbed phosphorus release rate in sediments (s^{-1}) 5 10^{-4};

α_{por} equal to unity in the water column and equal to $(1-p_{or})/p_{or}$ in sediments;

α_s (mg/l H_2S)/(mg/l C_d) 0.88;

α_μ (mg/l O2)/(mg/l $C_{al, ph, m}$) 2.66;

$\beta_{1,}$ β_2 : stochiometric constant;

$\gamma = K_{\gamma 1}C_{al} + K_{\gamma 2} C_{ph} + K_{\gamma 3} C_m + K_{\gamma 4}C_p$ light extinction factor;

μ_{cral}, μ_{crph}, μ_{crm} : vegetal growth rates;

μ_d : aerobic mineralization rate (s^{-1}) 6 10^{-6}, 5 10^{-5}, 3 10^{-6} ;

ν_s : settling velocity of particulate organic carbon (m/s) = 1 10^{-6};

ν_t : turbulent diffusion coefficient and dispersion coefficient in sediments;

ω_y, ω_d, ω_m, ω_b : yearly, daily, tidal and breeze speed periods (d) 365,(h)24,12,24.

BIBLIOGRAPHY

FRANCESCO CIOFFI is PHD in Environmental Engineering (Politecnico di Milano). In 1997 he became Associate Professor in Hydraulic Engineering at
Dipartimento di Idraulica,Traporti e Strade, Rome University 'La Sapienza'.
E-mail: francesco.cioffi@uniroma1.it

GIOVANNI CANNATA is PHD in Hydraulic Engineering. He is research assistant at the group of environmental hydraulic of Dipartimento di Idraulica, Traporti e Strade, Rome University 'La Sapienza'.
E-mail: giovanni.cannata@libero.it

MODELLING THE FIGHT AGAINST FOREST FIRES BY MEANS OF A NUMERICAL BATTLEFIELD

Yves Dumond
Laboratoire LISTIC
University of Savoie
Campus Scientifique
F-73376 Le Bourget-du-Lac Cedex
E-mail: Yves.Dumond@univ-savoie.fr

KEYWORDS
Business process modelling, geographic information system, 3-D visualization, numerical battlefield, tactical situation, forest fires, Asphodèle.

ABSTRACT

In this paper we describe the main features of Asphodèle, a software system that has been designed for the management of material and human resources deployed to fight forest fires. First, we present the procedures specific to this kind of operation, i.e. the technical framework in which French firemen operate. This leads us to focus on the cartographic data onto which the intervention strategies can be specified. These descriptions require the use of a set of specific graphical features. Then, the management of the resources involved in the operations is discussed as well as the predictions which are made concerning the spread of fire. At the end, the implementation of the system and possible future works are touched upon.

INTRODUCTION

In many parts of the world, forest fires are among the worse natural disasters. As a matter of fact, this bane can modify ecosystems resulting in the extinction of animal and vegetable species. Thus, it has always been a cause for concern in the Mediterranean basin, and because of climate change, many countries in Central and Northern Europe are now threatened. Therefore, coping with this problem is of great importance and depends both on prevention and action. Interventions against forest fires can involve up to several thousand firemen and hundreds of vehicles including aircraft. In spite of this, very few software environments are specifically dedicated to this purpose. Asphodèle, on the other hand, has been developed in close cooperation with a fire department in the South of France. Consequently, it faithfully reproduces this department’s operating methods. In particular, it enables the elaboration of a numerical battlefield called a “tactical situation”. From a technical point of view, Asphodèle is supported by a geographical information system (GIS) of which it uses the cartographic data. A specific editor then allows the capture of dedicated graphical elements and places them onto the maps provided by the GIS.

INTERVENTION MANAGEMENT

The virtual battlefield is a well-known concept in military applications. Indeed, this approach is especially suitable when the chain of command has to face very complex problems related to the management of numerous means. Therefore, very sophisticated environments, including multimodal interfaces (Baker and Stein 1992), have been implemented for that purpose. Such an approach can be very well adapted to the framework of the staff headquarters of fire departments. However, in France, most of the interventions against forest fires are managed from a mobile command post, i.e. a vehicle specifically equipped for this mission (figure 1).

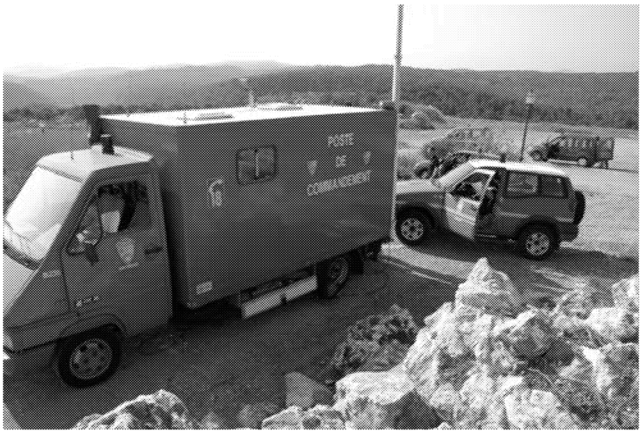


Figure 1 Mobile command post © JG.Bouillon / SDIS 06

The embedded technology mainly consists of:

1. Radio communication: by this means, orders are transmitted to the fire-fighters, who in turn keep the officers of the command post informed of the situation on the ground.

2. One or two portable computers connected to an additional 19-inch flat screen, as shown in figure 2.

3. If possible, the command post is connected to the telephone network in order to allow Internet communication with the regional staff headquarters.

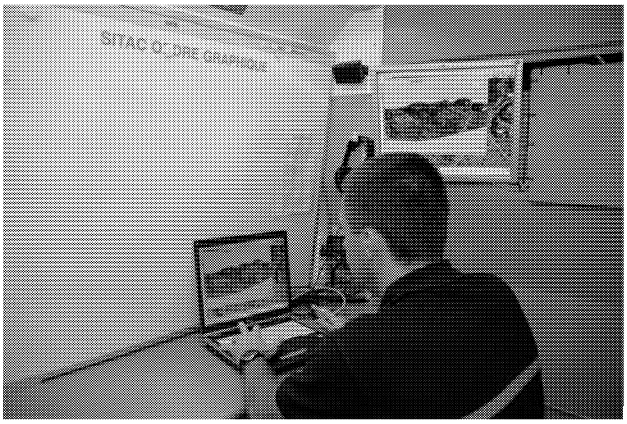


Figure 2 Portable computer in a mobile command post © JG.Bouillon / SDIS 06

So, the challenge we had to face was the implementation of an embedded numerical battlefield with the quite limited means described above. Following the example of military software, we intended to offer a very precise synthesis of the situation on the field, in particular information such as:

1. The geographic characteristics of the operating zones and the presence of any kind of man-made structure.

2. All the information regarding the fire, especially the related areas and the different spreading axes.

3. The location of all the resources engaged, i.e. heavy vehicles and fire-fighting units.

4. The specifics of all the actions performed, both past and still in progress.

Moreover, the system had to allow both 2-D and 3-D visualization. Starting from these observations, it was clear that a GIS should be the cornerstone of the software to be developed. Indeed, these systems provide a large amount of geographic data. Furthermore, they generally offer reusable program libraries for the implementation of dedicated add-on software.

GEOGRAPHICAL DATA

At the threshold of an intervention, the first task to be performed consists of an in-depth analysis of the operating zones. Numerous characteristics of the latter must be brought to the fore. Some of them are, by nature, present on the maps. Among these, we note:

1. The relief and its possible influence over the spread of fire.

2. Moreover, some areas may be hard to reach and this can restrict the means that can be used to fight the fire: the accessibility to the different sites is in fact conditioned by the road network.

3. The level and the nature of urbanization that can lead to locate areas that must be protected first.

Furthermore, other specific elements must be considered, namely:

1. The course of the high voltage lines that can be a serious danger to aircraft.

2. The layout of forest tracks that can make sites accessible.

3. The location of fire hydrants and of water tanks previously dispatched and filled in winter time.

By plotting this information on geological survey maps (figure 3), one obtains a set of base maps. These are available under different scales and serve as base data for the management of the interventions.

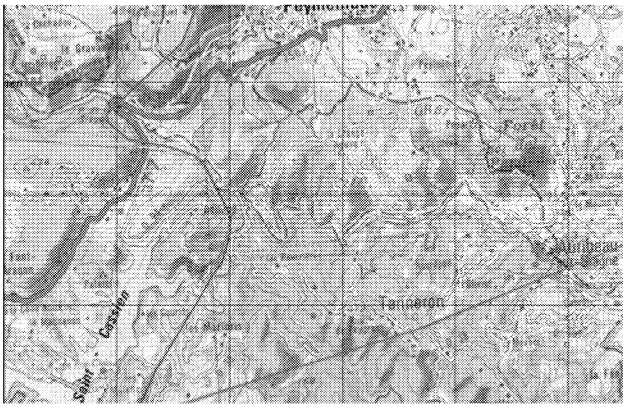


Figure 3 Base map © SDIS 06

Moreover, aerial photographs (figure 4) are also available and can be used in order to reach complementary objectives. For instance, they provide information such as the nature of the vegetation which is an important element in fire spread predictions.

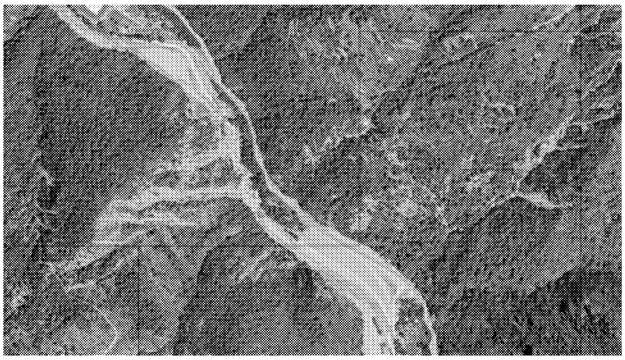


Figure 4 Aerial photograph © SDIS 06

Base maps and aerial photographs (from now on all referred to as “background maps” or more simply “maps”) offer thirteen different kinds of visualization of the operating zones. Each of them aims to bring to the fore a specific view: crests and thalwegs, planimetry, water resources, road network, winds, etc.

TACTICAL SITUATIONS

Specifying an intervention precisely consists in adding dedicated graphical items to background maps. These items are in conformity with a national standard that defines their form and their semantics. Thus, they are part and parcel of the firemen's professional culture. From an applicative point of view, this symbology refers to four different kinds of concepts:

1. The attributes of the fire: starting point(s), areas covered, different spreading axes, speed, etc.

2. Fixed means, for instance: water resources, hospitals, heliports, filling stations, parking areas, etc.

3. Mobile means, for instance: tankers, oil supply vehicles, logistics, mobile headquarters, medical units, psychological support, veterinary units, etc.

4. The actions performed by the units deployed on the ground, for instance: flank attacks whose goal is to reduce the width of the fire area, support lines along which the fire is supposed to be stopped, etc.

From a graphical point of view, the dedicated items (see figure 5 for some examples) can be divided into two sets:

1. Thirty different icons that can be laid down, without any modification, onto the maps by means of a simple drag-and-drop procedure. These icons can also represent fire attributes (fire starting point), mobile means (heavy vehicle columns), actions (water dropping by helicopter) or fixed means (water tank).

2. Twenty different kinds of lines whose drawing is performed by means of the mouse pointer or possibly a graphic tablet. These can either be straight linear (fire spread axes), surface (fire area) or curved structures (support line, flank attack).

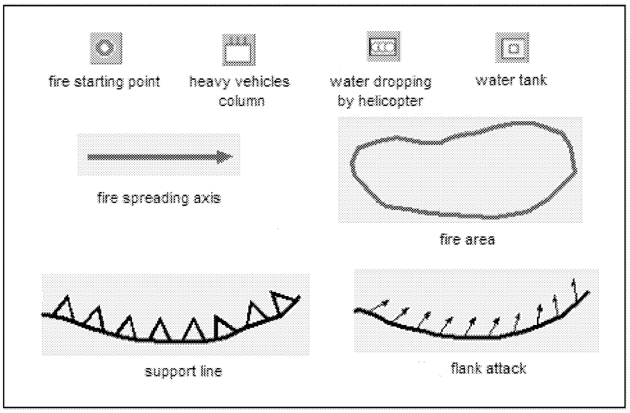


Figure 5 Graphical items

The capture of graphical items is achieved by means of a specific editor which is the heart of the system interface. This also manages the display of background maps.

Thus, the insertion of graphical items onto background maps leads to the creation of dedicated types of diagrams called "tactical situations" (figure 6). These are, in the full sense of the word, a *modelling* of the interventions insofar as they offer an abstract and synthetic view of the situation. Let us also note that the data required for their elaboration are provided by observers dispatched on the ground.

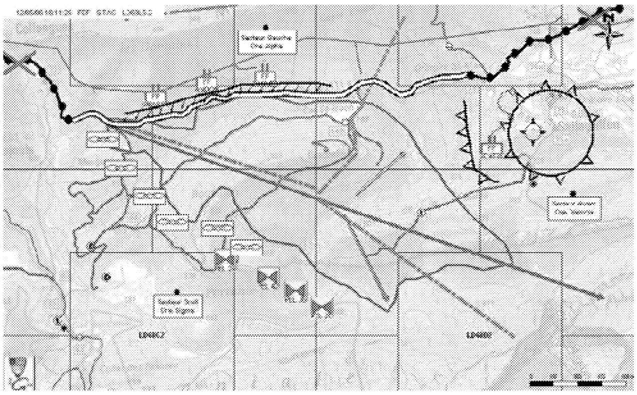


Figure 6 Tactical situation © SDIS 06

MOBILE RESOURCE MANAGEMENT

Mobile resources involved in the operations follow a standard life cycle: fire fighting units are successively required, obtained, available, engaged and finally relieved. Thus, a specific module of the system, the mobile resource manager (MRM), provides an overall view of all the mobile resources deployed on the ground. Roughly speaking, this may be regarded as a dedicated spreadsheet in which columns denote the successive steps of the life cycle detailed above. Furthermore, some complementary information, such as the specific equipment, the radio band, the duration of engagement, etc. allows an optimal management of each means.

Another important aspect in the management of interventions is the concept of division into sectors. A tactical situation can be divided into up to three sectors that can be managed separately. To each of these sectors is associated a column in the MRM. The latter works in interaction with the related tactical situation: when an item is transferred from one sector to another, the corresponding columns are automatically updated in the MRM.

PREDICTIONS ABOUT THE SPREAD OF FIRE

The prospective step is a key concept in the management of the interventions. This is especially important in requesting new resources as well as in relieving units at the right time. So, one would expect to make use of simulation techniques to predict the spread of fire. Indeed, research works are currently being carried out on this subject (Muzy et alt. 2005) and some systems such as BEHAVE (Rabner et alt. 2001) or FARSITE (Finney 1998) are in use in North America. Obviously, such an approach would be a complementary step to the one presented in this paper. However, it remains difficult to put into practice for several reasons:

1. The amount of data to be processed is considerable. Among these, we must consider: the relief, the (generally heterogeneous) nature of the vegetation and its water content, the level of urbanization, the force and the direction of the winds, etc.

2. Unforeseeable events, such as the projection of burning pine cones, can greatly modify the spread of fire.

3. The results obtained by the firemen involved in the fight cannot be predicted and consequently cannot be considered in the simulation model.

4. Relevant simulation models could probably not be run on the portable computers available in mobile headquarters.

Thus, the actual practical impact of simulation techniques remains, to date, quite limited. In order to cope with this problem, the task of fire spread prediction is entrusted to experienced officers. The following are the tools provided by Asphodèle in order to support this activity:

1. 3-D visualization (figure 7) of tactical situations for all types of maps, i.e. base maps or aerial photographs.

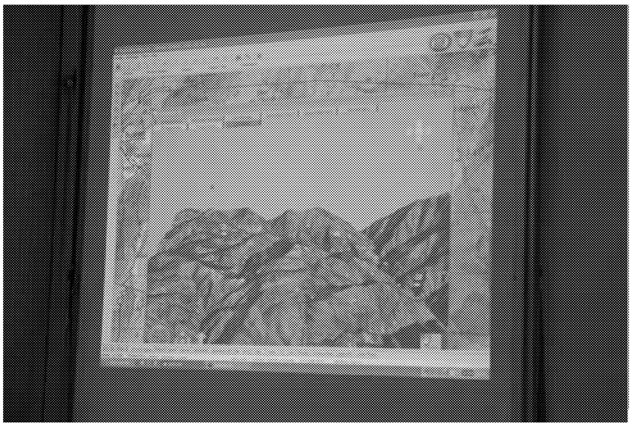


Figure 7 3-D visualization © JG.Bouillon / SDIS 06

2. Aerial photographs reflect, by nature, the actual state of the ground, which is not the case for the other kinds of maps.

3. Specific maps contain the areas covered by the fires that have occurred in the same zone during the past decade. These provide very interesting information about the possible scenarios.

SYSTEM IMPLEMENTATION

Asphodèle has been implemented under Windows by means of the programming toolkit of the SIG GeoConcept (GeoConcept 2005). The system is built from a set of C++ classes, and its architecture (figure 8) consists of three different Windows tasks:

1. The *interface,* that handles the interactions both with the user and the SIG GeoConcept. In particular, maps display mechanisms include scrolling and scale variation. Furthermore, when they are captured, graphical items are immediately inserted in the GeoConcept database. By this means, they are part of the tactical situation until they are removed. The interface also includes a MySQL database that duplicates all the data the tactical situation is made up of, especially those present in the MRM. Then, once they have been encoded in the form of an XML file, this information can easily be sent through Internet, for instance to the regional staff headquarters.

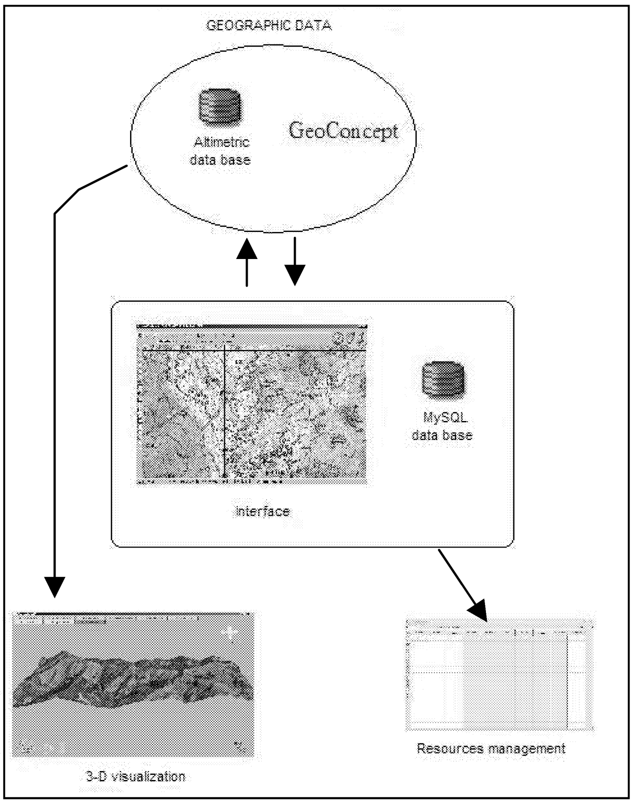


Figure 8 Asphodèle architecture

2. The *MRM* holds all the information about mobile means present in the tactical situation. From a technical point of view, the MRM is synchronised with the interface by means of sockets. Moreover, we may note that the interface and the MRM can be run on two different computers: in such a case, one

operator is dedicated to the elaboration of the tactical situation while another manages the mobile means.

3. The 3-D visualization task has been implemented with the OpenGL environment (Shreiner 2005). It provides all the standard functionalities of 3-D visualization: rotations, zooming, compass, etc. Moreover, the relief is deduced from an altimetric database.

CONCLUSION

Asphodèle is now in use in wide areas of the South of France. Although the numerical battlefield is not a new concept, at least two lessons can be learned from this project:

1. Firstly, we should note that GIS turned out to be extremely powerful environments for the implementation of software systems involving geographical information. Therefore, carrying out Asphodèle without one of these environments would have been totally impossible.

2. Although other software systems dedicated to tactical situation management are available, Asphodèle's main characteristic is, according to the users, its ergonomics. In point of fact, this system has been developed in permanent contact with experienced firemen and consequently it reflects their operating methods. To a certain extent, it can therefore be regarded as a business process-oriented software. This approach is probably the only one that makes a system like Asphodèle satisfactory. Indeed, during the interventions, nervous strain, if not stress, is a permanent psychological constraint. Thus, the software system should not impose principles different from the firemen's way of thinking.

Furthermore, future works may include:

1. The use of formal specification techniques for the description of the properties of the system. This question appears to be of great importance if we consider the underlying human and material stakes. Concurrent computing models such as pi-calculus (Milner 1999) or Linda (Gelernter 1985) may therefore be suitable for this purpose.

2. The implementation of an expert system to support the activity of fire spread prediction: this would probably lead to modelling qualitative reasoning. However, skills seem to be hard to synthesize in that case.

3. An increase in the ability to operate in a communicating environment: for instance, with the proviso that GPS transmitters would be available on vehicles, the corresponding items could automatically be inserted at their correct location in tactical situations with a high level of reliability. Furthermore, the use of wireless communications would make possible a distributed version of Asphodèle, with computers present both in staff and mobile headquarters as well as in the vehicles deployed on the ground. The consecutive rise in information flow would then be a determining advantage in the management of the interventions.

REFERENCES

Baker, M.P. and R.J. Stein. 1992 "The CAVE: Audio-Visual Experience Automatic Environment". *Communications of the ACM,* Vol.35, No 6, 65-72.

Finney, M.A. 1998. "FARSITE: Fire Area Simulator-model development and evaluation". RMRS – Research paper 4, Ogden, UT: US Department of Agriculture, Forest Service, Rocky Mountain Research Station.

Gelernter, D. 1985. "Generative communication in Linda". *ACM Transactions on Programming Languages and Systems,* Vol.7, No 1, 80-112.

GeoConcept SA, 2005. *Geoconcept, The Geographical Information System,* Reference Guide.

Milner, R. 1999. Communicating and Mobile Systems: The pi-calculus, 1st edition. Cambridge University Press, Cambridge, England.

Muzy, A.; E. Innocenti; G. Wainer; A. Aïello; J.F. Santucci; and P.A. Santoni. 2005. "Specification of discrete event models for fire spreading". *Simulation: transactions of the society of modelling and simulation international,* No 81, 103-117.

Rabner, K. W.; J.P. Dwyer; and B.E. Cutter. 2001. "Fuel model selection for BEHAVE in midwestern oak savannas". In *Northern Journal of Applied Forestry,* Vol.18, No.3, 74-80. Jamestown, ND: Northern Prairie Wildlife Research Center Online.

Shreiner, D.; M. Woo; J. Neider; and T. Davis. 2005. OpenGL Programming Guide: the official Guide to Learning OpenGL, 5th edition, Addison-Wesley Professional, Boston, MA. USA.

ACKNOWLEDGEMENTS

The author wishes to thank:

1. The Fire Department of the Alpes-Maritimes (SDIS 06) for its support during the Asphodèle project.

2. Mrs Eryl-Anne Baylis and Mr Antony Wright for their careful review of this paper. Their comments have been a great help in improving the English style of the text.

BIOGRAPHY

YVES DUMOND obtained his PhD in Computer Science from the University of Nice Sophia-Antipolis in 1988. He is currently associate professor at the University of Savoie (France), where he has worked since 1990. In particular, he has been involved in the Eurêka project PVS 2434. His topics of interest include formal specifications, concurrency theory, especially process calculi, and software engineering.

WEB
BASED
SIMULATION

INTEGRATION OF WEB BASED SIMULATORS IN THE SINPL PLATFORM[§]

Alberto Coen-Porisini
Ignazio Gallo
Antonella Zanzi

Dipartimento di Informatica e Comunicazione
Università degli Studi dell'Insubria
Via Mazzini 5, 21100 Varese, Italy
E-mail: {alberto.coenporisini|ignazio.gallo|antonella.zanzi}@uninsubria.it

KEYWORDS
Simulators integration, Web-based distributed simulation, Web services, open-source software.

ABSTRACT

In the last years the interest in software applications accessible through the World Wide Web is increased and the field of simulation has been influenced as well as many others. In particular, Web-based simulation may be viewed as the natural evolution of distributed simulation.
SINPL (Simulator Integration Platform) is an open-source software platform aimed at supporting the simulation design and the integration of heterogeneous simulators in a distributed environment. This paper discusses the use of the Web services technology in the SINPL platform, both to obtain Web-based simulators and to implement a Web-based simulation environment.

INTRODUCTION

In the field of simulation, existing distributed architectures provide different levels of abstraction, and each of them offers services aimed at fulfilling different needs.

The High Level Architecture (HLA) (Kuhl et al. 2000), an IEEE standard, defines an approach to integrate autonomous simulators into a single distributed simulation system. Simulations are described in terms of federations of federates, where a federation is a simulation system composed of two or more simulator federates communicating through the Run-Time Infrastructure (RTI). HLA defines a common architecture supporting reuse and interoperability of simulations and is intended to have a wide applicability to many different areas. However, its practical use requires highly skilled people because of its inherent complexity. Moreover, HLA does not fully address the problem of providing an integrated design environment.
This feature is instead addressed in SINPL (Coen-Porisini et al. 2004), an open-source software platform for supporting design and simulation activities, and allowing the integration of existing heterogeneous discrete event simulators in a distributed environment.

In the last years the interest in software application accessible through the World Wide Web (Web in the following) is increased and also the field of simulation was influenced. With the large utilization of Web applications, proposals for Web-based access to simulation programs are increasing (Kuljis and Paul 2000). In particular, Web-based simulation may be viewed as the natural evolution for ubiquitous distributed simulation.

This paper discusses the use of the Web services technology in the SINPL platform, both to obtain Web-based simulators and to implement a Web-based simulation environment.
The paper is organized in the following way: first of all works in the Web-based simulation field are briefly reviewed; then the SINPL platform is introduced, followed by the description of the Web services used in the platform itself; finally, some conclusions are drawn.

RELATED WORK

Research and development efforts on Web-based simulation include server side simulations, client side simulations, and distributed Web-based simulators. The first two approaches allow one to access already existing simulation environments through the Web, the third one consists of an extension of the distributed simulation architecture to Web-based network infrastructures.

In server side simulations, it is possible to remotely access the simulation program from a Web browser, execute it on a server platform and see the results on the browser itself. Typically, these simulations use Common Gateway Interface (CGI) scripts or Java Servlets and they allow the user to provide the parameters of model execution using the Web browser. In these approaches a single instance of the simulation runs on a server and passes the results to the invoking client.

Client side applications allow users to run a simulation model on their desktop, where they provide input parameters using the Web browser. Typically, the simulation model and the user interface run on the client side as applet (Java program embedded in a Web page). In this case, the simulation executes on the client rather than on the server.

There are several Java-based simulation libraries that permit the creation of simulation programs as Java applications and applets. Among these are JSIM (Miller et al. 2000a), JavaSim (Little 2001), SimJava (McNab and Howell 1996), DEVSJAVA (Sarjoughian and Zeigler 1998) and Simkit (Buss and Stork 1996).

[§] Work supported by the Italian MIUR-FIRB – Tecnologie abilitanti per la Società della conoscenza ICT.

Researches were done also in the field of distributed simulations on the Web, with components running on different machines. Multiple users can interconnect with the same underlying simulation model through Web browsers from different locations.

Different technologies are investigated to find an ideal substrate for Web-based simulation, for example Remote Method Invocation (RMI) (Page et al. 1997) and component technologies such as Enterprise Java Beans (EJB) and Jini (Miller et al. 2000b).

In the area of distributed simulation the relationship between the High Level Architecture and Web-based simulation has also been investigated (Page 1998). Efforts, for example, were done in order to implement Web-based federated simulation systems using Jini (Huang and Miller 2001) or CORBA (D'Ambrogio 2004).

A different approach for the distributed simulation on the Web is represented by the use of Web services. Synergy between simulation and Web services is twofold: simulation is used to study Web services composition (Chandrasekaran et al. 2002) and Web services are used to build simulation environments. Some examples of the latter are: a framework called XMSF (Pullen et al. 2004) supporting interoperation among different software systems, including HLA federations, on heterogeneous platforms; a Web services system for managing scientific simulation metadata (Hawick and James 2005); in the computational fluid mechanics area, a system for doing multiphysics simulation of a coupled fluid, thermal, and mechanical fracture problem (Chew et al. 2003).

THE SINPL PLATFORM

SINPL is an open-source software platform that allows one to carry out distributed simulation. The platform allows one to integrate existing heterogeneous discrete event simulators in a distributed environment. The platform is composed by a set of tools supporting the different activities that one has to carry out when designing (and then enacting) a distributed simulation. Roughly speaking, the tools composing the SINPL platform allow one to:

1. create an Information Model, that is to define the basic components characterizing a given application domain, such as flexible manufacturing systems, telecom systems, etc. The tool allowing users to define Information Models is called InfoCreator;
2. create a Simulation Architecture, that is to design a distributed architecture composed of instances of the components defined in the Information Model. When designing a Simulation Architecture one has to define how many components he/she wants to use and how they are connected one to another. The tool allowing users to define Simulation Architectures is called SED (Simulation EDitor);
3. execute a Simulation Architecture, that is allowing the components (simulators) to exchange data among them in order to carry out a distributed simulation.

In this paper we focus on the execution support provided by the platform, while a detailed presentation of the tools composing the platform along with the presentation of the approach supported by the SINPL platform can be found in (Coen-Porisini et al. 2004).

The tool in charge of the execution of the simulation, called *Distributed Simulation Controller* (DSC), coordinates the simulation execution among the distributed software components by managing the communication among the different simulators. The DSC determines how data should be passed by one simulator to another by using the information provided by the Simulation Architecture, which can be viewed as a High Level Petri Net (HLPN) (Jensen 1997). The HLPN provides information on the control flow among simulators and it is used by the DSC to actually control the distributed simulation. More specifically, the DSC allows the simulators to exchange events and provides mechanisms to ensure a correct synchronization among the simulators. The interested reader may refer to (Carullo et al. 2006) for an in-depth discussion of the synchronization issues in SINPL.

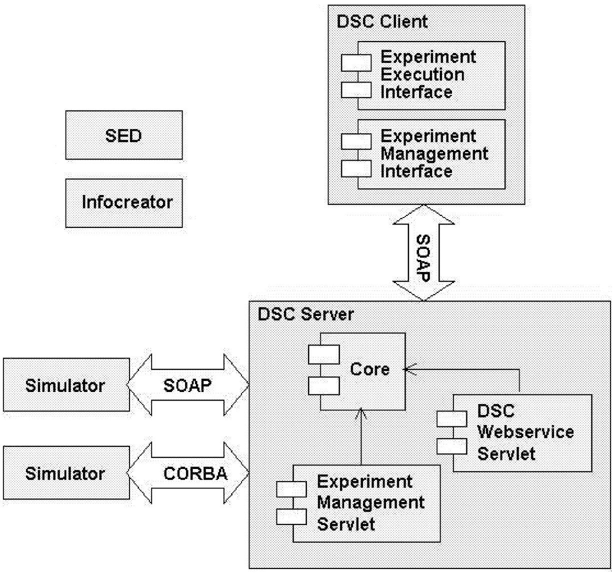


Figure 1: SINPL – Deployment diagram

SINPL allows the communications among heterogeneous simulators by requiring that they all conform to a given interface defined by SINPL. Thus, the way in which a simulator can be integrated in SINPL is by developing a software adapter to provide a simulator with the interface required by SINPL. The communication between the DSC and the simulators (and their adapters), was originally based on CORBA, which is a middleware defined by the OMG for allowing distributed heterogeneous objects to communicate. However, one of the drawback of using CORBA is that it might be difficult to access remote objects (i.e., simulators) when their location is outside the LAN where the DSC is installed. Thus, we decided to provide the DSC with a second way of communication based on the usual Web protocols that are used, such as HTTP. As a result, the simulators can be viewed as Web applications. Moreover, the DSC is based on a client-server architecture. The server side is in charge of controlling the distributed simulation, while the client side allows users to control, monitor and/or review the distributed simulation itself.

Figure 1 shows the deployment diagram of the platform (with the Web services extensions too). All the tools have been developed using Java and are open-source.

In the next section we discuss the way in which the Web technology has been integrated in the SINPL platform.

SINPL AND THE WEB

A Web service is an accessible software component deployed on the Web. Such component is described by an interface (described in a format such as WSDL – Web Service Description Language) listing the operations that are supported.

Web services support XML-based distributed computing using SOAP (Simple Object Access Protocol), a lightweight protocol for exchanging structured information (XML-based messages) in a decentralized, distributed environment. It consists of three parts: an envelope defining a framework for describing what is in a message and how to process it, a set of encoding rules to express instances of application data types, and a convention for representing remote procedure calls and replies. SOAP can be used through different protocols like FTP, SMTP and HTTP.

The HTTP protocol is particularly used as it works well with Internet infrastructure. More specifically, SOAP can usually work through network firewalls without requiring changes to the firewall filtering rules. This is a major advantage over other distributed protocols, which are normally filtered out by firewalls. Other advantages are the interoperability between various software applications running on different operating systems and the reuse of services and components within an infrastructure. Moreover, Web services are loosely coupled thereby facilitating a distributed approach to application integration.

From the performance point of view, because of the XML format, SOAP is slower than other middleware technologies, such as CORBA, or binary network protocols, such as RMI or IIOP. However, this may not be an issue if only small messages are used.

In what follows we discuss how Web services have been integrated in the SINPL platform. More specifically, such technology has been used in three different ways: first of all to allow simulators to communicate among them by means of the DSC; secondly to implement the Web SINPL Manager application devoted to managing the DSC; finally to allow the DSC server to communicate with the DSC client(s).

Web-based Simulators

As said before, simulators can exchange data using either CORBA or SOAP as communication protocol. Since communication occurring among simulators is actually controlled by the DSC, each simulator sends the data it produces to the DSC that, in turn, determines to which simulators the data received have to be sent. In order to handle two different communication mechanisms the DSC has to abstract away from the actual communication mechanism (SOAP or CORBA).

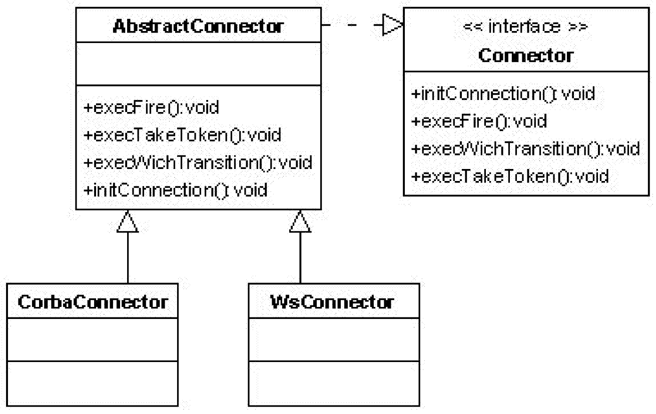


Figure 2: The Design Pattern Strategy

Thus, the DSC is implemented by using the design pattern Strategy has shown in figure 2. More specifically, the DSC contains a module, implemented by a Java interface named *Connector*, providing all the methods needed for sending data to simulators. The interface connector is then implemented by an abstract class, whose name is *AbstractConnector*, providing the implementation of the methods that are independent from the communication mechanism. Finally, class *AbstractConnector* is extended by two concrete classes (*CorbaConnector* and *WsConnector*) implementing the methods that depend on the communication mechanism.

When a simulator has to send data to the DSC, it must use a callaback interface provided by the DSC itself. Thus in order to allow simulators to use two different communication mechanisms, the DSC provides two different callback interfaces, one for CORBA and one for SOAP, implemented by means of two different Java packages. Figure 3 shows the structure of the package named *dsc.callback.callbackws*, which provides the implementation of a Web service that simulators can use to send data to the DSC.

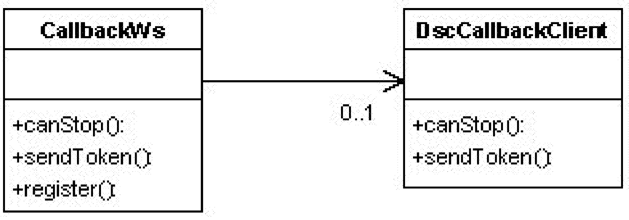


Figure 3: The *dsc.callback.callbackws* package

Notice that such Web service runs as an independent process with respect to DSC. Thus, the actual communication occurs in the following way: the simulator sends the data to the callback Web service, which, in turn, using a standard interprocess communication mechanism, connects to the DSC for sending the data.

Referring to the class diagram of Figure 3, class *CallbackWs* contains the methods provided to the simulators as Web

service, while *DSCCallbackClient* represents the class used to exchange data with the DSC.
Finally, using the tools provided by the development environment used (Apache Axis), class *CallbackWs* was automatically transformed in order to comply with the WSDL standard and the needed proxies were generated.

DSC WebService and Web SINPL Manager

The DSC WebService (implemented as a Servlet) is the interface that is used both to allow the remote management of the DSC and to obtain information about running simulations. In particular, it is possible to start a new distributed simulation, stop a running one, and obtain the list of the currently running simulations. Notice that the execution of more than one distributed simulation at the same time is permitted. The DSC WebService is used by the DSC client (see later on).

The Web SINPL Manager application allows the remote management of simulations setup and of users' accounts. This Web application has a graphical interface (figure 4) allowing the remote utilization of the DSC functionalities for setting up a simulation and the download of all the platform tools (InfoCreator, SED, and DSC Client) which one has to install on his/her PC in order to make use of the platform.
From the Web SINPL Manager user's interface is also possible to both upload data for a new simulation experiment and execute a simulation experiment already available.
Before the start of a simulation, all the involved simulators have to be active and, when the communication between the simulators and the DSC is based on CORBA, the ORB has to be in execution as well.
The Web SINPL Manager was implemented in the DSC server as a Servlet in the same Web application that contain the DSC WebService Servlet. For the implementation of the Web services the package Apache Axis was used.

DSC Client

The main goal of this Web application is the "animated" visualization (available in 2D and 3D) of a simulation. In the views the connections among simulators are shown with data sent on each connection.
The graphical interface of the DSC Client (implemented as a Java SWING application) shows both 2D and 3D representation of the simulators as depicted in figure 5.
The user can either monitor an on-going simulation or review an already executed simulation since all the relevant events are logged by the DSC. In this way one can examine all the messages exchanged by the simulators and possibly replay the simulation itself starting from any time instant.

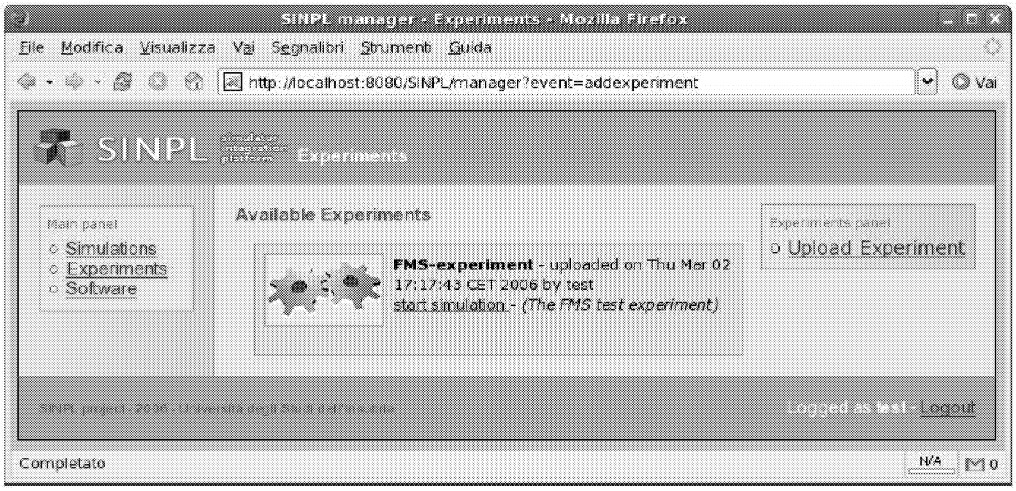
More specifically, the DSC client provides a summary of the number of events exchanged in each time instant in which the simulation was executing (timestamp list window), the list of events actually exchanged in a given time instant (events at timestamp window), all the information associated with each event (event details and data sent windows) along with the 2D and 3D animated representation of the on-going simulation.

CONCLUSIONS

In the present work the application of the Web services technology in the SINPL platform was introduced. In particular, SOAP was added as another communication channel (to the already existing one based on CORBA) between the main platform module (DSC) and the simulators. In this way a distributed Web simulation becomes feasible, deploying entities on different remote sites, without the technical difficulties that could arise with CORBA ORBs.
Future work regards the development of a "white pages" tool for Web simulators in order to find in a simple way the available simulators on the Web.

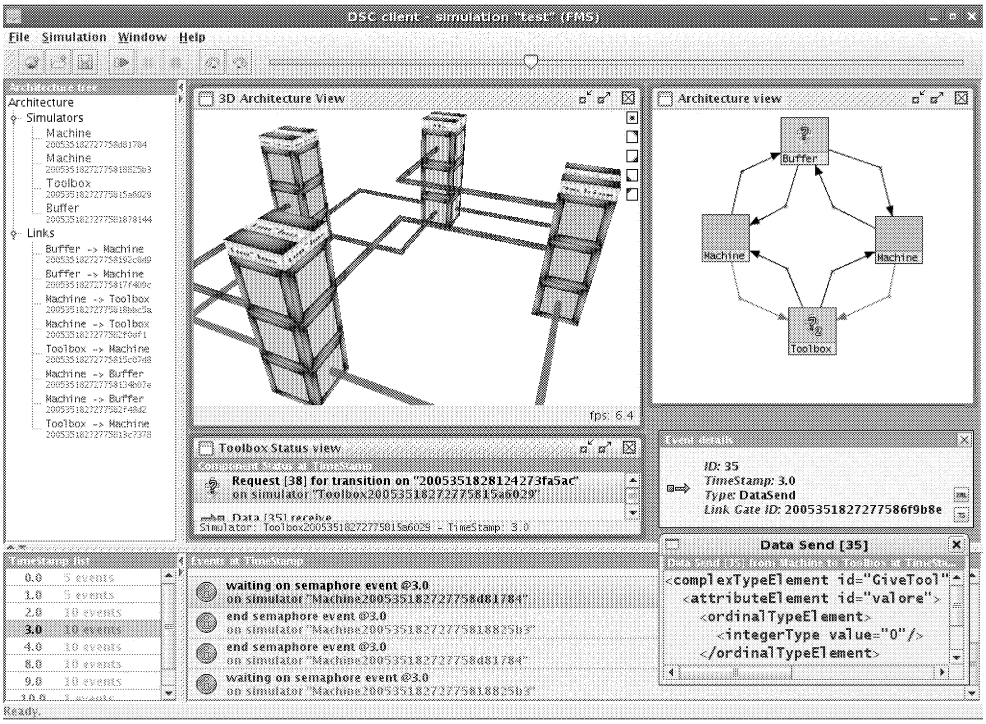
REFERENCES

Buss, A.H. and Stork, K.A. 1996. "Simulation on the World Wide Web Using Java." In *Proceedings of the 1996 Winter Simulation Conference* (Coronado, CA, December 8-11 1996), 780-785.


SINPL manager - Experiments - Mozilla Firefox
File Modifica Visualizza Vai Segnalibri Strumenti Guida
http://localhost:8080/SINPL/manager?event=addexperiment
Vai
SINPL
Experiments
Main panel
Simulations
Experiments
Software
Available Experiments
Experiments panel
Upload Experiment
Logged as test - Logout
Completato
N/A
0

FMS-experiment - uploaded on Thu Mar 02 17:17:43 CET 2006 by test
start simulation - *(The FMS test experiment)*

Figure 4: Web SINPL Manager Interface



Request [38] for transition on "2005351828124273fa5ac"
on simulator "Toolbox200535182727758l5a6029"

Data [35] receive

Simulator: Toolbox200535182727758l5a6029 - TimeStamp: 3.0

ID: 35
TimeStamp: 3.0
Type: DataSend
Link Gate ID: 200535182727758 6f9b8e

0.0 5 events
1.0 5 events
2.0 10 events
3.0 10 events
4.0 10 events
8.0 10 events
9.0 10 events
10.0 3 events

waiting on semaphore event @3.0
on simulator "Machine200535182727758d81784"

end semaphore event @3.0
on simulator "Machine20053518272775818825b3"

end semaphore event @3.0
on simulator "Machine200535182727758d81784"

waiting on semaphore event @3.0
on simulator "Machine20053518272775818825b3"

```
<complexTypeElement id="GiveTool"
  <attributeElement id="valore">
    <ordinalTypeElement>
      <integerType value="0"/>
    </ordinalTypeElement>
```

Figure 5: Simulation Execution Graphical Interface (DSC Client)

Carullo, M.; Zanzi A.; Gallo, I; and Coen-Porisini, A. 2006. “An events synchronization approach for integration of simulators in a distributed environment”. In *Proceeding of Industrial Simulation Conference – ISC 2006* (Palermo, Italy, June 5-8 2006), 74-78.

Chandrasekaran, S. ; Silver, G ; Miller, A.J.; Cardoso, J.; and Sheth, A.P. 2002. “Web service thechnologies and their synergy with simulation.” In *Proceedings of the 2002 Winter Simulation Conference* (San Diego, CA, December 8-11 2002), 606-615.

Chew, L.P; Chrisochoides N.; Gopalsamy, S.; Gerd Heber G.; Ingraffea, A.R.; Luke, E.; Neto, J.B.C.; Pingali, K; Shih, A.; Soni, B.K; Stodghill, P.; Thompson, D.; Vavasis, S.A.; and Wawrzynek, P.A. 2003. “Computational Science Simulations Based on Web Services.” *International Conference on Computational Science*, Springer-Verlag, 299–308.

Coen-Porisini, A.; Gallo, I.; and Zanzi A. 2004. “Designing and enacting simulations using distributed components.” In *Computer and Information Sciences – ISCIS 2004. Proceedings of the 19th International Symposium* (Kemer-Antalya, Turkey, October 27-29 2004). Springer, 706-717.

D’Ambrogio, D.G. 2004. “Using CORBA to enhance HLA interoperability in distributed and web-based simulation.” In *Computer and Information Sciences - ISCIS 2004. Proceedings of the 19th International Symposium* (Kemer-Antalya, Turkey, October 27-29 2004). Springer, 696-705.

Hawick K.A. and James H.A. 2005. “Web services for remote management of scientific simulation.” In *Proceeding International Conference on Web Technologies, Applications, and Services* (Calgary, Canada, July 4-6, 2005), M. H. Hamza (Ed.), 100-105.

Huang, X and Miller, J.A. 2001. “Building a Web-Based Federated Simulation System with Jini and XML.” *Annual Simulation Symposium 2001* (Seattle, WA, April 22-26 2001). IEEE Computer Society, 143-150.

Jensen, J. 1997. “Coloured Petri Nets. Basic Concepts, Analysis Methods and Practical Use.” Vol. 2, Analysis Methods, Monographs in Theoretical Computer Science, Springer-Verlag (2nd corrected printing).

Kuhl, F; Weatherly, R.; and Dahmann, J. 2000. “Creating computer simulation systems – An Introduction to the High Level Architecture.” Prentice Hall PTR.

Kuljis, J. and Paul, R.J. 2000. “A review of web based simulation: whither we wander?” In *Proceedings of the 2000 Winter Simulation Conference* (Orlando, FL, December 10-13 2000). ACM, 1872-1881.

Little, M.C. 2001. “JavaSim User's Guide. Public Release 0.3, Version 1.0.” University of Newcastle upon Tyne.

McNab, R. and Howell, F.W. 1996. “Using Java for Discrete Event Simulation.” In *Proceedings of the Twelfth UK Computer and Telecommunications Performance Engineering Workshop*, (University of Edinburgh, UK), 219-228.

Miller, J.A; Seila, A.F.; and Xiang X. 2000a. "The JSIM Web-Based Simulation Environment," *Future Generation Computer Systems (FGCS),* Special Issue on Web-Based Modeling and Simulation, Vol. 17, No. 2 (October 2000). Elsevier North-Holland, 119-133.

Miller, J.A; Seila, A.F.; and Tao, J. 2000b. “Finding substrate for federated components on the web.” In *Proceedings of the 2000 Winter Simulation Conference*, (Orlando, FL, December 10-13 2000). ACM, 1849-1854.

Page E.H.; Moose, R.L.; Sean, J.; and Griffin P. 1997. “Web-based simulation in SimJava using remote method invocation.” In *Proceedings of the 1997 Winter Simulation Conference.* (Atlanta, GA, December 7-10 1997). ACM, 468-474.

Page E.H. 1998. “The rise of web-based simulation: implications for high level architecture.” In *Proceedings of the 1998 Winter Simulation Conference* (Washington DC December 13-16 1998). ACM, 1663-1668.

Pullen, J.M; Brunton, R.; Brutzman, D.; Drake, D.; Hieb, M.; Morse, K.L.; and Tolk, A. 2004. “Using web services to integrate heterogeneous simulations in a grid environment”. In *Proceedings of International Conference on Computer Science 2004* (Krakow, Poland, June 2004). Elsevier, 99-106.

Sarjoughian, H.S. and Zeigler, B.P. 1998. “DEVSJAVA: Basis for a DEVS-based collaborative M&Senvironments.” In *Proceedings of the 1998 International Conference on Web-Based Modeling & Simulation* (San Diego, Ca, January 11-14 1998), 29-35.

GROUPSIM: EXTENDING A SIMULATION GROUPWARE TO ALLOW INTEROPERABILITY

Celso M. Hirata; Tony Calleri França; and Vakulathil Abdurahiman
Divisão de CIência da Computação
Instituto Tecnológico de Aeronáutica
S.J.Campos – SP – Brazil
e-mail: hirata@ita.br

Germano de Souza Kienbaum
Laboratório Associado de Computação e Matemática Aplicada
Instituto Nacional de Pesquisas Espaciais
S.J.Campos – SP – Brazil

KEYWORDS

Discrete Event Simulation, Web Services, Web-based Simulation, Groupware, Interoperability

ABSTRACT

The simulation process involves the collaboration of different participants such as analysts, programmers, statisticians, and simulation software operators. GroupSim is a cooperative environment to help the construction of simulation software using the WWW platform. Although GroupSim promotes user collaboration, it does not inter-operate with other system. The present work proposes the extension of GroupSim by using the Web Services technology with ACD models described in XML to allow interoperability. The Web Services include functionalities to share models and their data.

INTRODUCTION

The simulation process involves the collaboration of different roles participants such as analysts, programmers, statisticians, and simulation software operators. Many tasks of the simulation process require the collaboration of such participants. The participants are generally distributed geographically and need to move (travel) and meet in order to perform the tasks. Some of the tasks resort to collaborative tools. Collaborative tools have been used to reduce the number of face-to-face meetings. These tools include video-conference system, conference call, Messenger, etc. Groupware systems support groups of people engaged in a task or common objective and provide an interface for a collaborative computing environment (Ellis et al. 1991). GroupSim (Aráujo Filho et al. 2004) is an Internet based groupware to help the construction of simulation software. The construction of simulation software involves modeling, programming, and some tasks of experimentation. Groupsim uses the concepts of distributed modeling with automatic program generation, and distributed control of experimentation. GroupSim promotes user collaboration and uses tightly coupled architecture, however, it does not inter-operate with other system in any level. The reasons for a tightly coupled architecture are related to the provision of simultaneity and awareness. The simultaneity is related to the notion of actions occurring and being perceived instantaneously. *Awareness* is the property of illusion of presence that a user has of other users sharing the same workplace. A description of awareness can be found in (Greeenberg et al. 1997). So, due to its nature, groupware systems are not concerned with interoperability.

Web service is a software system designed to support interoperable machine-to-machine interaction over a network. It usually refers to those services that use SOAP-formatted XML envelopes and have their interfaces described by WSDL. Comparing to Java RMI and CORBA, web services are seen as loosely coupled reusable software components that semantically encapsulate discrete functionality (Stencil Group 2003) whose primary goal is interoperability.

The present work proposes the extension of GroupSim to allow interoperability. The inter-operability is achieved by using the Web Services technology and XML representations of ACD models. The Web Services extended GroupSim to store and share models and their input data.

In the next section, we review GroupSim and Section 3 presents the interoperable extension of GroupSim. Section 4 presents some related work and Section 5 ends with the conclusions and some comments.

GROUPSIM

GroupSim is a collaborative environment for discrete event simulation that works on the World Wide Web through simple Java-enabled browsers. A discussion of how groupware systems like Groupsim can be used in the simulation processes and a description of GroupSim can be found in (Araujo Filho et al. 2004). GroupSim allows not only the construction of the model but also its execution. More specifically, GroupSim allows collaborative model editing with automatic code generation and collaborative control of experiments. It is argued that these two tasks certainly need cooperation between collaborators. The modeling task requires the interaction between the simulation analyst and the client while the experimentation control might require actions of simulation analyst, statistician, and the client during the verification of the model.

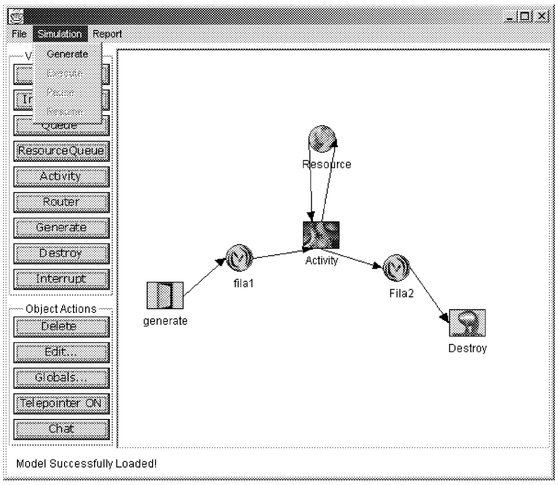
For construction, the extended ACD is adopted. The Active Cycle Diagrams (ACD) (Pidd 1990) is a modeling notation for discrete event simulation similar to Timed Petri Nets. Some extensions to ACD were proposed by Hirata & Paul (Hirata and Paul 1996) and are adopted in GroupSim. In GroupSim, users interact themselves through collaborative graph editor of ACD. The ACD representations are automatically translated to Java objects. This takes the manual programming task out of the loop. To accomplish this, GroupSim uses the mapping of the extended notation to Java classes, which compose the simulation program proposed by (Araujo Filho and Hirata 2004).

The distributed control of experimentation allows users to control the execution of an experiment. Once a user starts the execution, any other (remote) user can pause or stop, and afterwards start or resume the execution. When the execution is over, a report containing all the relevant information, as specified during the modeling, can be easily generated.

GroupSim uses a collaborative editor to construct the ACD. The editor is an artifact-presentation object pair. It contains information shared by the clients, which is the ACD and the presentation object, which is the graphic interface that shows the ACD. The shared model is composed of two types of data: the graphic data and the configuration data. Configuration data includes all the information that describes the simulation model, which is used to generate the simulation program.

GroupSim uses the client-server paradigm. There is a server application that manages the client applications. The application server allows clients to be added, work, and leave the groupware. The application server also keeps all the data including the models that are being constructed. The client application is used by the collaborator and allows the interaction. GroupSim uses the replicated consistency strategy. All data except the configuration data are replicated in all clients.

Figure 1 shows the main interface of GroupSim. In this figure a simple ACD is being edited. Elements can be added to the model by clicking on the element in "Visual Objects Bar" and dragging it to the work area. If an element is selected it can be edited or deleted. To remove an element, all arcs related to the element must be deleted first – this helps to keep consistency of the model. A *popup menu* named *Simulation*, located in the top area of the window, allows both generating the program and controlling the execution of the simulation. The control is made by selecting *execute*, *pause* or *resume* option. To start the program, the generation must have completed with success. Operations of storage and load of model (in bytecodes) is made on the local disk if the mode of operation is local otherwise, i.e. the mode of operation is cooperative, the model is stored on the server.



Model Successfully Loaded!

Figure 1: Interface of Model Cooperative Editor.

GroupSim allows having multiple models being edited by different groups of users or by just one user in stand alone mode. After a *login*, the user can choose the edit room that he wants to enter, and afterwards, he enters in the edit environment. In a room, the user only interacts with users of that room. Figure 2 shows a window where the room "ModelServer2" is being selected.

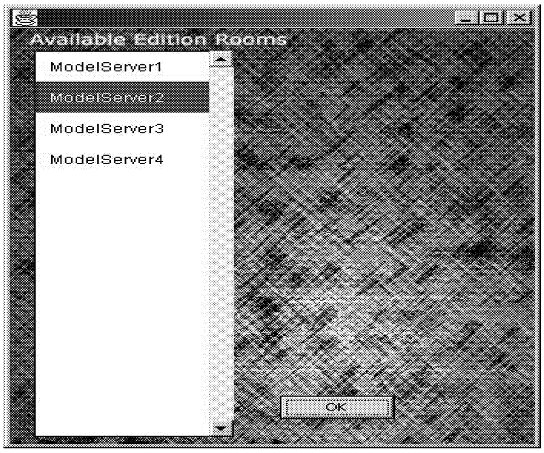
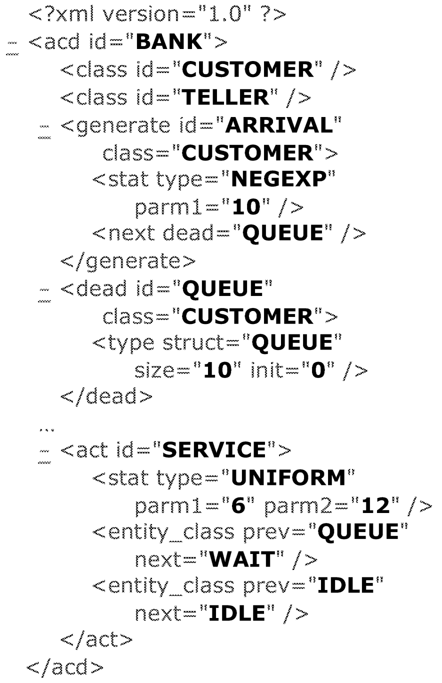


Figure 2: Window that allows the edition rooms.

The basic functionalities that are standard and useful in groupware include *Chat* and *Telepointer*. The chat functionality allows exchanging messages among users connected in the room.

A desirable functionality is the inter-operability with other systems. In order to meet this goal, it is necessary to use a common specification for ACD models and the specification should be exchangeable between systems. So the XACDML (Gil and Hirata 2003) is adopted and employed in GroupSim. In Figure 3, we illustrate part of an XACDML representation of the bank system with 3 tellers and a single queue.



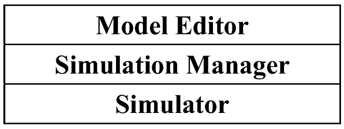
```
  <?xml version="1.0" ?>
- <acd id="BANK">
     <class id="CUSTOMER" />
     <class id="TELLER" />
   - <generate id="ARRIVAL"
          class="CUSTOMER">
        <stat type="NEGEXP"
             parm1="10" />
        <next dead="QUEUE" />
     </generate>
   - <dead id="QUEUE"
          class="CUSTOMER">
        <type struct="QUEUE"
             size="10" init="0" />
     </dead>
   ...
   - <act id="SERVICE">
        <stat type="UNIFORM"
             parm1="6" parm2="12" />
        <entity_class prev="QUEUE"
             next="WAIT" />
        <entity_class prev="IDLE"
             next="IDLE" />
     </act>
  </acd>
```

Figure 3. Some XACDML code

Through *popup menu* named *File*, located in the top area of the main window in Figure 1, the user has options to open as well as save XACDML files. In cooperative mode, the XACDML file is stored in the file system of the application server otherwise it is stored in the file system of the client.

GROUPSIM WITH WEB SERVICES

A Web Service is a software system design to support the inter-operability between machines through a network. It contains an interface described in a format that can be processed by a machine. The format is the WSDL (Web Services Description Language). Other systems interact with Web services as prescribed by its description through SOAP messages. SOAP messages are also XML representation and are transmitted using the HTTP protocol with XML serialization. The goal of a Web Service is to provide functionality in the name of an owner – user or organization - of the service. The provider is responsible for listing, implementing, and maintaining the service. The requester is a user or organization that makes uses of the service. For this interaction, both parties must agree on the semantics and mechanics of the message exchange. The mechanics is documented in the description of the Web Service made in WSDL. The semantics of the message exchanged is the contract between the parties. The contract governs the mechanics of the interaction and despite of its importance, it is not explicitly registered in the system as a protocol.



Model Editor

Simulation Manager

Simulator

Figure 4: Architecture of GroupSim

Figure 4 shows the three layers used by GroupSim. The Model Editor layer includes the graphical representation of the model. The Simulation Manager layer is responsible to instantiate the objects and construct the simulation program. The Simulator layer manages the simulation execution.

GroupSim is cooperative tool so that it is possible to execute just locally in the client or in a client-server style. When GroupSim runs in the client-server style, the client application connects to the server, which takes over the activities of the layers Simulation Manager and Simulator. The client application realizes the activities of the Model Editor layer. The client application communicates with the server application using RMI (Remote Method Invocation). The server application makes available remote methods corresponding to the operations that are made in the model. The client application makes available remote methods so that the server can notify the client of other updates of the model, which were made by other clients (users). Through this mechanism, the user can become aware of the changes and intentions of other users.

The methods of remote interface of the server include all methods that the client can call. They are: methods to register as user, leave the server obtain list of users; methods to obtain/release locks to edit the model; methods to publish new position of telepointer; methods to edit and visualize and retrieve the model, add/remove/list states and arcs, and retrieve/modify parameters of objects; methods for chat implementation; methods that encapsulate the access to the Web Service WSInputSim; methods to control the execution of the simulation: generate the program, start/pause/resume the execution; methods to persist data: save/load model in binary format, save reports of the observers.The interface ModelClient includes all the methods whereby the server communicates with the client. The methods are called by the server in order to notify the user of some state change triggered by some action of other user.

WSInputSim is the Web Service to store and retrieve the information of the simulation tasks, namely the data input and the models in XACDML. WSInputSim is implemented in EJB (Enterprise Java Beans), which is hosted in the Web application server BEA Weblogic 6.1. It uses the bridge JDBC-ODBC to connect to a relational database, which was implemented in MS-Access. Figure 5 shows the two-layer architecture of Web Service WSInputSim.

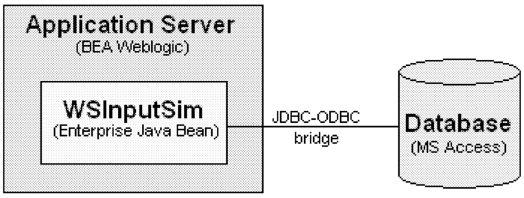


Figure 5: Two-layer architecture of Web Service WSInputSim

WSInputSim implements na interface with six method, which allow to list, save, and load input data sets of model; and list, save, and load models specified in XACDML;

The methods of WSInputSim allow to: (1) create an input data set - it requires a name related to the model; (2) return a list (array) of names (String) of InputSet related to a model; (3) return an array de objects Input that correspond to the given InputSet; (4) list the names of all the existing ACD models in the database; (5) returns a String with XACDML code corresponding to the model specified by a given name; and (6) create an ACD model ACD with a given name.

The database used by WSInputSim is composed of three tables, which are shown in Figure 6.

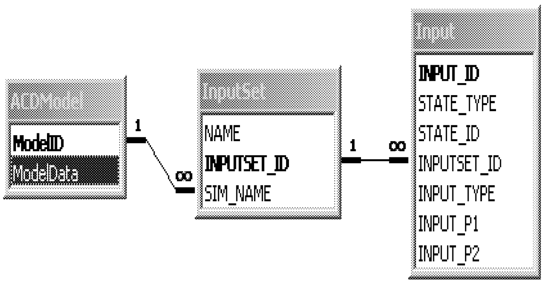
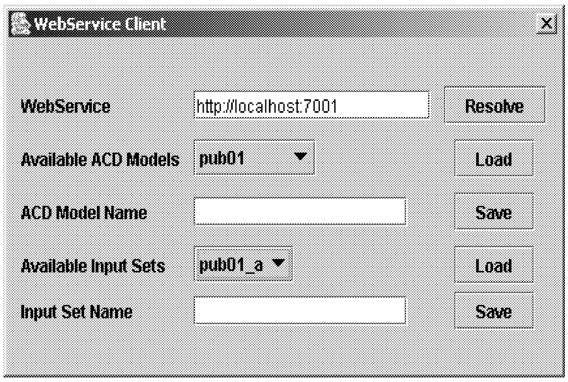


Figure 6: Entity relationship model of WSInputSim

The table *ACDModel* contains records that describe models. Each model can use many input data set. The input data set is described in the table *InputSet.* Each input data set can use many state descriptions. The state description basically is described in the table Input and it contains the following information: state type (active, resource, and queue), queue discipline, maximum size of the queue, type of probabilistic distribution and parameters, and so on.

GroupSim uses WSInputSim to obtain a list of models and load a specific model. Once the model is loaded, the users can collaborate and edit the model. Once the modeling is complete, users of GroupSim can generate and execute the program. After the execution the results are produced. Instead of using its own data, users of GroupSim can call WSInputSim to list the data sets that can be used by that model, and choose one of the data sets, and execute the simulation program again.



WebService http://localhost:7001

Available ACD Models pub01

ACD Model Name

Available Input Sets pub01_a

Input Set Name

Figure 7: Window to call WSInputSim from GroupSim

Figure 7 shows the window that allows accessing WSInputSim. The button Resolve loads the dropdown menus *Available ACD Models* and *Available Input Sets* with names of the ACD models and input data sets respectively. The resolution of input data set is performed only if an ACD model is selected. Once the model or data set is selected it can be loaded in GroupSim. Once the model or data set is loaded it can be saved in the database managed by WSInputSim.

An interface between GroupSim and WSInputSim named InputSimClient was developed. The interface allows accessing the database in both modes local and cooperative. Figure 8 illustrates the possible interactions using the interface.

RELATED WORK

Henriksen et al (Henriksen et al. 2002) propose the Web Based Simulation Center (WBSC). WBSC is a framework to support Web based simulation cooperative work on simulation projects. As an ASP based solution, however, WBSC neither provides for synchronized cooperative modeling nor for interoperability.

Kilgore (Kilgore 2002) shows how web services with .NET technology can be used in simulation. He proposes the creation of a web services for simulation model development. The web service is created to access static information during the execution of the simulation. However the proposal does not include groupware.

Chandrasekaran et al (Chandrasekaran et al. 2002) discuss how web services can be used in the simulation process. Basically, they discuss three possibilities:

models as web services, environmental components as web services, and model federates as web services. In our approach, we implemented the second possibility with the extension of a groupware.

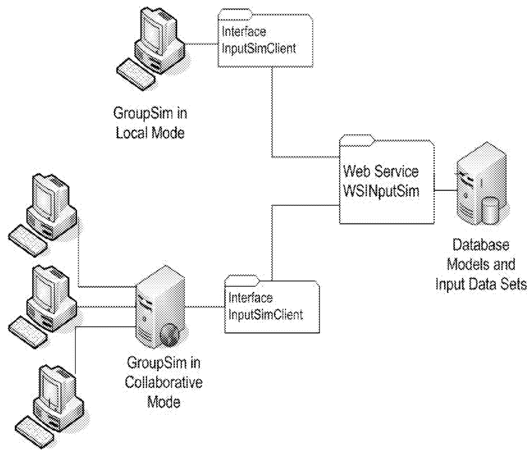


Figure 8: Accessing WSInputSim in Local or Collaborative Mode.

CONCLUSIONS AND COMMENTS

The work reported here is a result of the evolution of GroupSim along the last years. Now, GroupSim is capable of generating and storing models specified in XACDML, retrieving and building the objects (program) from XACDML, and lastly exchanging models and their data using Web Services.

The proposal of enabling GroupSim to resort to services from other systems needs to be validated in practice. The real benefit will come only if the organizations start to exchange discrete event simulation models, besides this move needs a pragmatic approach and some work still need to be made. First, GroupSim is not secure when it accesses information in other systems. Other issues include ownership of models, permissions to retrieve models, permission to store and publish them, and so on.

Some future work includes enhancements on the capability of GroupSim. An immediate work is to enable GroupSim to retrieve encapsulated sub-models instead the whole model. Other work is the development of other Web Services. For instance, web services to generate and run simulation programs using XACDML models.

REFERENCES

ELLIS, C. A.; GIBBS, S. J.; REIN, G. *Groupware: Some issues and experience*, Comm. ACM, v. 34, n. 4, p. 38-58, Jan. 1991.

ARAUJO FILHO W.L.; HIRATA, C.M.; and YANO, E.T. GroupSim: a collaborative environment for discrete event simulation software development for the WWW Transactions of the SCS. Vol. 80 No. 6 June 2004 pp 257-272. ISSN 0037-5497.

GREEENBERG, S., GUTWIN and ROSEMAN. *Collaborative Interfaces for the Web – Human Factors and Web Development*. Edited by Chris Forsythe, Eric Grose and Julie Ratneer, LEA Press, 1997.

THE STENCIL GROUP. *Defining Web Services*. Date of access: May 30, 2003. http://www.stencilgroup.com/ideas_scope_200106wsdefined.html.

PIDD, M. Computer Simulation in Management Science, 1990, Wiley.

HIRATA, C. M. and PAUL, R. J. *Object-Oriented Programming Architecture for Simulation Modeling*, International Journal in Computer Simulation, v. 6, n. 2, p. 269-287, 1996.

ARAUJO FILHO, W.L and HIRATA, C.M. Translating Activity Cycle Diagrams to Java Simulation Programs, Proceedings of the 37th Annual Simulation Symposium, IEEE, p. 157, 2004.

GIL, J.N. and HIRATA, C.M. *XACDML – Extensible ACD Markup Language*. Proceedings of the 36th Annual Simulation Symposium, IEEE, p. 343-350, 2003.

HENRIKSEN, J.O.; LORENZ, P.; HANISCH, A.; OSTERBURG, S.; SCHRIBER, T.J.; *Web based simulation center: professional support for simulation projects*. Proceedings of the 2002 Winter Simulation Conference. E. Yucesan, C.-H. Chen, J.L.Snowdon, and J.M.Charnes, eds., p. 807-815, 2002.

KILGORE, R. A. *Simulation web services with .NET technologies*. Proceedings of the 2002 Winter Simulation Conference. E. Yucesan, C.-H. Chen, J.L.Snowdon, and J.M.Charnes, eds.. p. 841-846, 2002.

CHANDRASEKARAN, S.; SILVER, G.; MILLER, J.A.; CARDOSO, J.; and SHETH, A. P. *Web Service technologies and their synergy with simulation.* Proceedings of the 2002 Winter Simulation Conference. E. Yucesan, C.-H. Chen, J.L.Snowdon, and J.M.Charnes, eds, p. 606-615, 2002.

BIOGRAPHY

CELSO M. HIRATA is an associate professor at the Computer Science Department at Instituto Tecnológico de Aeronáutica (ITA). He obtained his Engineer title (1982) and his MSc (1987) at ITA. He obtained his PhD (1995) from Imperial College (UK). His interests are in Discrete Event Simulation, Distributed Processing, and Software Project Management.

AGENT BASED SIMULATION IN BIOLOGY

ANALYSIS OF THE RELATIVE IMPORTANCE OF THE HUMORAL VERSUS THE CELLULAR RESPONSE DURING THE ACUTE STAGE OF HIV INFECTION: RESULTS FROM MULTI-AGENT COMPUTER SIMULATIONS.

Ashley Callaghan
Heather J. Ruskin,
Ray Walshe
Dublin City University, Dublin 9,
Ireland
{acallaghan, hruskin, rwalshe}@computing.dcu.ie

ABSTRACT

Results of multi-agent simulations of the immune response to HIV during the acute stage of HIV infection are presented here. The model successfully recreates the viral dynamics associated with the acute phase of infection i.e. a rapid rise in viral load followed by a sharp decline to what is often referred to as a 'set point' as a result of the joint response of the humoral and cellular arms of the immune system. The results obtained from our simulations indicate that the relative strength of the cellular response is the key factor that determines the variation found in different individuals' response to infection with HIV.

1. INTRODUCTION

In this paper results obtained from recent work carried out to model the immune response during the acute stage of HIV infection by means of a multi-agent computational mode are presented. Although recent research has realised a better understanding of HIV, (for a review of the last twenty years of research see Stevenson (2003)), the exact mechanisms by which HIV causes AIDS remain unclear. The question being addressed within this paper, is to what extent may the relative strength of the humoral versus the cellular response and vice-versa, be responsible for the variation seen in different individuals' response to HIV infection. The remainder of this paper is laid out as follows: section 2 gives a brief overview of the immune response to HIV and discusses how HIV manages to survive this response. Section 3 introduces computational models of the immune system. Section 4 gives an overview of the model used in this research, and is followed in section 5 by some of the results obtained and explains their significance. The final section of the paper concludes and offers some perspectives on future work planned.

2. IMMUNE RESPONSE TO HIV

The immune response to HIV infection is characterised by a complex web of interactions involving numerous cells and molecules of the host's defense system. What follows is a brief outline of some of the key processes as described, Benjamini *et al* (2000) and Leffell *et al* (1997) and references therein.

Infection with HIV presents the immune system with two basic problems. The first involves the removal of virions from the blood stream, and the second involves the destruction of infected cells.

2.1 Removal of virus from the blood stream

Removal of virus (antigen) from the blood stream involves antigen presenting cells (APC's), T Helper cells (CD4) and antibodies. APC's include dendritic cells, macrophages and B cells. The role of these cells in HIV infection is to ingest any free virions they encounter. Before ingestion can take place the APC must first bind to one of the epitopes of the antigen, (epitope is a specific molecular configuration found on the antigen surface). Each of the approximately 10^{10} B cells found in the human immune system presents a practically unique antibody receptor (BCR) that allows it to bind with high affinity to a specific set of antigenic epitopes. Macrophages and dendritic cells in comparison are equipped with non-specific receptors resulting in a much lower affinity for a particular antigen than a well-matched B cell. The advantage is that macrophages and dendritic cells, can respond to a wide range of antigens, which is of particular importance in the early stages of infection, where the number of well-matched B cells is typically very low. After ingesting the virion the cell breaks it down into small fragments known as peptides. The peptides are then transported to the surface of the cell, where they are presented to T Helper cells, by means of the Major Histocompatibility Complex class 2 (MHC2). Just like a B cell, a T Helper cell also contains a specific receptor (TCR) on its surface. However, unlike B cells they do not use their receptors to recognize and bind to a foreign antigen. Instead they attempt to bind to MHC2-peptide complexes that may be present on the surface of B cells. Upon successful binding, complex chemical signals are released, and these signals stimulate both cells to begin several rounds of division, in the process creating clones of cells that are well equipped to deal with the antigen. The resulting B cells can be one of two types. Long lived memory B cells that will take part in future immune responses if the same antigen is encountered again, or short-lived plasma B cells (PLB's), whose purpose is to secrete huge amounts of antibodies. Antibodies are complex organic molecules known as immunoglobulins, which can bind to, inactivate and help remove antigens.

2.2 Destruction of infected cells

HIV preferentially infects CD4 T cells, but also has the ability to infect other cell types in particular macrophages and dendritic cells (Stevenson (2003)). After entering the target cell, HIV remains in a latent state until favourable host cell factors trigger the beginning of transcription, (Bailey *et al* (2004). Once transcription begins, HIV replicates at a phenomenal rate, with the infected cell producing a vast number of new virions in a short space of time, (Perelson et al (1996), Perelson and Nelson (1999)). As newly formed virions are assembled they bud from the surface of the infected cell and are released into the peripheral blood, (Greene and Peterlin (2002)), following which, they themselves can infect other target cells. The constant strain on the cell membrane of the infected cell that results, generally leads to the infected cell bursting and the release of any virions it contains, (Fauci (1988)). The key entities by which the immune system attempts to combat this process are T Killer cells, (also referred to as CD8 T cells), and the Major Histocompatibility Complex class 1 (MHC1) complexes. Like T Helper cells, T Killer cells also possess a TCR. However, instead of using it to bind to an MHC2-peptide complex, they instead attempt to bind to MHC1-peptide complexes. Once transcription begins, the infected cell attempts to present HIV peptides to T Killer cells by means of a MHC1 complex. Upon successful binding the T Killer cell secretes chemical signals that result in the death of the infected cell, and also begins several rounds of division, which results in a clone of the cell, that can also recognise the antigen.

2.3 HIV manages to survive despite these responses

Although the immune system generally manages to bring the viral load, (level of HIV), to low levels within weeks to months of initial infection, HIV is never completely eliminated but instead remains present in low concentrations. Immune escape is usually attributed to two factors: its incredible rate of replication and high rate of mutation, (Perelson *et al* (1996), Perelson and Nelson (1999)). New strains of HIV constantly emerge during infection as a result of the error-prone nature of replication and the rapid turnover of virus in infected individuals (Coffin (1995)). These strains have the potential for immune escape, as the clones of cells that will have developed against the wild type virus, will generally lack the ability to respond to new strains of the virus (Derdeyn and Silvestri (2005)).

3. COMPUTATIONAL MODELLING OF THE IMMUNE SYSTEM

Cellular automata (CA) type models have been extensively applied to investigate numerous areas of the immune system (Pandey (1991), Celada and Seiden (1992), Zorzenon dos Santos and Coutinho (2001), Ruskin *et al* (2002)). CA have a number of attractive characteristics for modelling a system such as this. They correspond directly to the components and processes of interest in biological terms; this is important as it means that the approximations made in allowing the simulation to be carried out are usually more biological than mathematical in nature. This representation allows for an easier assessment of the applicability of results and allows direct interaction with the model in immunological terms rather than mathematical terms. CA also allow easy modification of the complexity of interactions without introducing new difficulties in solving the model, as non-linearities are not intrinsically difficult to deal with in these types of models. However, as with all models, there are limitations and disadvantages to attempting to model the immune system by means of cellular automata. The main limitation is that, due to the processing power and memory requirements required to scale up, only relatively small system size can be considered, compared to that of the actual immune system. Another disadvantage is that it can be very difficult to discover general laws about system behaviour.

3.1 Different types of Cellular Automata

Broadly speaking CA models as applied to the immune response can be described as one of three types, (i) physical space models, (ii) shape space models or (iii) physical-shape space hybrids. In a physical space model the interactions between the various entities are determined solely on their type and location. Examples of such models as applied to the immune system include (Pandey (1991); Mannion *et al* (2000); Ruskin *et al* (2002); Zorzenon dos Santos and Coutinho (2001)).). A significant limitation with these models results from the fact that no distinction is made between different entities belonging to the same class. In reality instances of the various entities such as the B cells and T cells are equipped with unique receptors on their surface, which allow them to bind preferentially but not exclusively to other complementary entities. By treating all instances of a particular entity as being identical, physical space models ignore the dynamics that emerge due to this important characteristic. Perelson and Oster (1979) introduced the concept of generalised shape space, as a means to represent the dynamics of antibody-antigen binding. The central idea behind the theory is that the features, which govern the behaviour of the receptors, can be represented by N integer parameters. By combining the N parameters into a vector, each receptor can be viewed as a point in N dimensional space, (Fig. 1a). Cells sharing the same receptor have the same N dimensional vector and reside at the same point in shape space. An important limitation, with regard to these models, is that they ignore the physical location of the various entities. Instead, the likelihood of two entities meeting in physical space is usually represented as a fixed probability (Hershberg *et al* (2001)). This is a problem as it ignores, the fact that the circulation pattern of immune cells is known to change upon detection of infection (Benjamin *et al* (2000*)*). A physical-shape space hybrid model is one that takes into account interactions with respect to both the physical location relative to each other of the cellular entities and their affinity (in shape space). They allow one to model the fact that behaviour of cellular entities depend both on its physical location in the body (physical space) and its affinity for the other entities (shape space). Examples of such models include, (Celada and Seiden (1992), Bernaschi and Castiglione (2001), Burns *et al* (2003, 2004a, 2004b)). An entity's location in shape space can be represented in a number of ways including a N dimensional integer vector as noted previously, while another popular way is by means of a bit-string. In this case, the receptors present on the surface of cellular entities are represented, by means of a binary string of length L. Two strings X and Y are said to be complementary to one another, at a given position, if a integer "one" on string X is matched by a "zero" on string Y and vice-versa. The number of positions at which the two strings are complementary is referred to as the Hamming distance, (Fig. 1b).

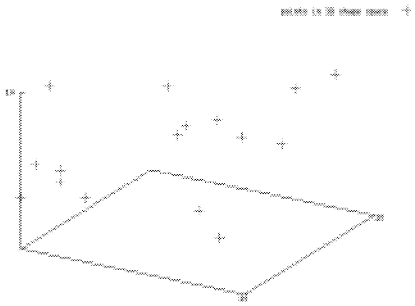


Fig. 1a: series of points in 3D shape space, where each dimension lies in the range0-10.

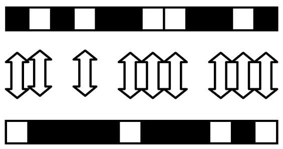


Fig. 1b: Two 12 bit-strings representing receptors, squares coloured black represent 1, while squares coloured white represent 0. Arrows indicate where the two strings are complementary. The Hamming distance in this case is 9

4. MULTI-AGENT MODEL: EXTENDING THE HYBRID PARADIGM

The Celada and Seiden model referred to as IMMSIM (1992), forms the basis for the work presented here, as it allows for the simulation of both the humoral and cellular arms of the human immune system in the presence of foreign antigens. The model simulates the interactions between a number of immune system entities on a two-dimensional hexagonal lattice, with periodic boundary conditions in both directions, which can be thought of as representing a single lymph node. The primary sources of lymphocytes i.e. the thymus and bone marrow are modelled separately: the thymus being implicitly represented by positive and negative selection of immature T cells before they enter circulation, while the bone marrow produces mature B cells.

IMMSIM can be thought of as being a physical-shape space hybrid, in so far as the interactions between entities depend both on their respective location on the lattice and also on physical characteristics (i.e. the receptors present on their cell surface). Bit strings are employed in the model to implement the shape-space concept, with the cell surface receptors of the various entities represented by means of a string consisting of N bits (Table 1 gives a list of the various properties represented by bit-strings for each of the entities represented in our model). For the simulations discussed here N was set to 12, which results in a potential repertoire of 2^{12} unique receptors. HIV is itself represented by two bit strings representing the epitope and peptide respectively, giving 2^{24} possible strains of the virus. Affinity between two entities is calculated as a function of the Hamming distance between two bit-strings (see Fig. 2).

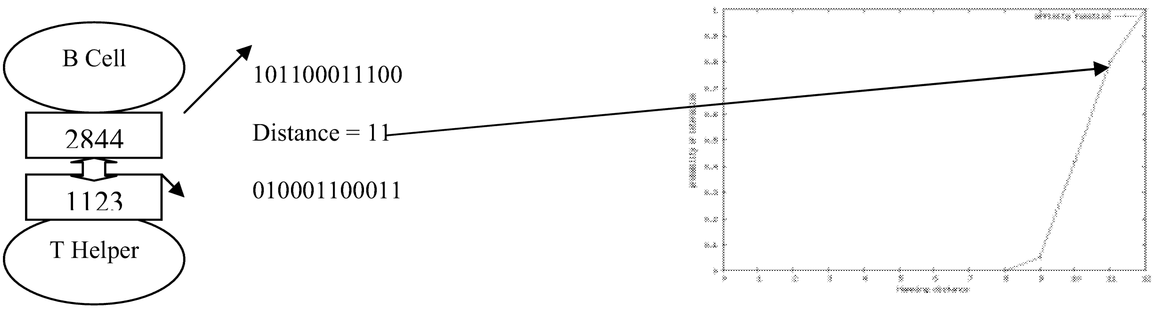


Fig. 2(a): depicts the interaction between a B Cell and a T Helper cell. The B cell is presenting a peptide on its MHC2 complex. The Hamming distance between the MHC2 complex on the surface of the B cell and the T cell receptor is found to be eleven

Unlike the vast majority of immunological models, the model of Celada and Seiden incorporates an additional level of complexity, in that it also represents the intracellular processes of digestion and presentation. The endogenous antigen is fragmented and combined with MHC1 molecules for presentation on the cell surface to T Killer (CD8) receptors, while the exogenous antigen is cut into small fragments known as peptides for presentation on the cell surface to T Helper (CD4) receptors. These processes are implemented in the model using the following steps; (i) the bit string representing the peptide is divided into two separate bit strings corresponding to the left and right hand sides of the peptide respectively, (ii) the Hamming distance between these strings and the right hand side of the bit string representing the MHC complex in question is then calculated, (iii) the probability of the peptide binding to the MHC complex for presentation on the cell surface is determined as a function of the greater of the two Hamming distances, (iv) if binding is successful the bit string corresponding to the peptide side in question is then combined with the left hand side of the MHC complex for presentation on the cell surface. This additional level of complexity, allows one to investigate factors, such as, to what extent do different individuals MHC complexes, influence their ability or lack thereof, to mount a successful response to different antigen.

Initial quantities for each of the cell types can be found in Table 1. The simulation proceeds in discrete time steps, each corresponding to eight hours of real life. At each time step the lattice sites are updated sequentially with the full set of interactions being executed in a random order; (in reality many of these interactions take place in parallel, and the introduction of a random element is an attempt to offset any bias introduced as a result of the sequential nature of computer programs). After the interactions have taken place, factors such as cell division, cell death and regeneration, growth of virus in infected cells, antigen presentation on MHC complexes etc., which result from the interactions taking place in the previous step, are implemented. Finally, at the end of each time step, entities can diffuse to adjacent lattice sites conditional on the number of entities currently present at the potential new location

Table 1**:** Initial quantity of each cell type

B	TH	TK	DC	MA	PLB	EP
260	875	435	350	350	0	350

Fig. 2(b) The Hamming Distance in 3(a) is used as input into this affinity function; which returns the probability of interaction which in this case is 0.8

Entity key:
B, lymphocyte B cell; TH, lymphocyte T Helper Cell or CD4; TK, lymphocyte T Killer cell or CD8; MA, Macrophage; DC, Dendritic cell; EP, epithelial cell; PLB, Plasma Secreting B Cell;

5. RESULTS

To investigate the relative importance of the humoral versus the cellular response and vice-versa, and their possible contribution with regard to different individuals showing a wide range of responses to HIV infection, numerous simulations were performed. The sole variation between the different runs was the bit string representing either the MHC1 or MHC2 complex. By varying the MHC1 complex we are affecting the ease with which the immune system can mount a successful cellular response whilst varying the MHC2 complex affects the ease with which an effective humoral response may be mounted. All other quantities and characteristics that define the artificial immune system, such as the various receptors found on the cell surfaces, the strain of HIV, and the initial quantities of each cell type were kept the same for all runs. As mentioned previously, in order for presentation to take place either the left or right hand side of the peptide must bind to the right hand side of the MHC complex in question. Since the length of the bit strings has been set to 12 for these simulations the distance between either side of the peptide and the right hand side of the MHC complex will be in the range 0-6 inclusive. By choosing a peptide whose right and left hand side bit strings are identical we can then chose seven different bit strings that can be used to represent the MHC complexes so that each one has a unique Hamming distance with respect to the peptide of the original ‘wild’ strain of the virus. In addition each of the seven bit strings has an identical left hand side therefore ensuring that the only variation over the different runs of the simulations is the ease with which the immune system can stimulate either the humoral or the cellular response as a result of its ability to successfully bind and present peptides on the MHC2 or MHC1 complex respectively. When varying the MHC1 complex the MHC2 complex is fixed to ensure an average humoral response while fixing the MHC1 complex ensures an average cellular response when we vary the MHC2 complex

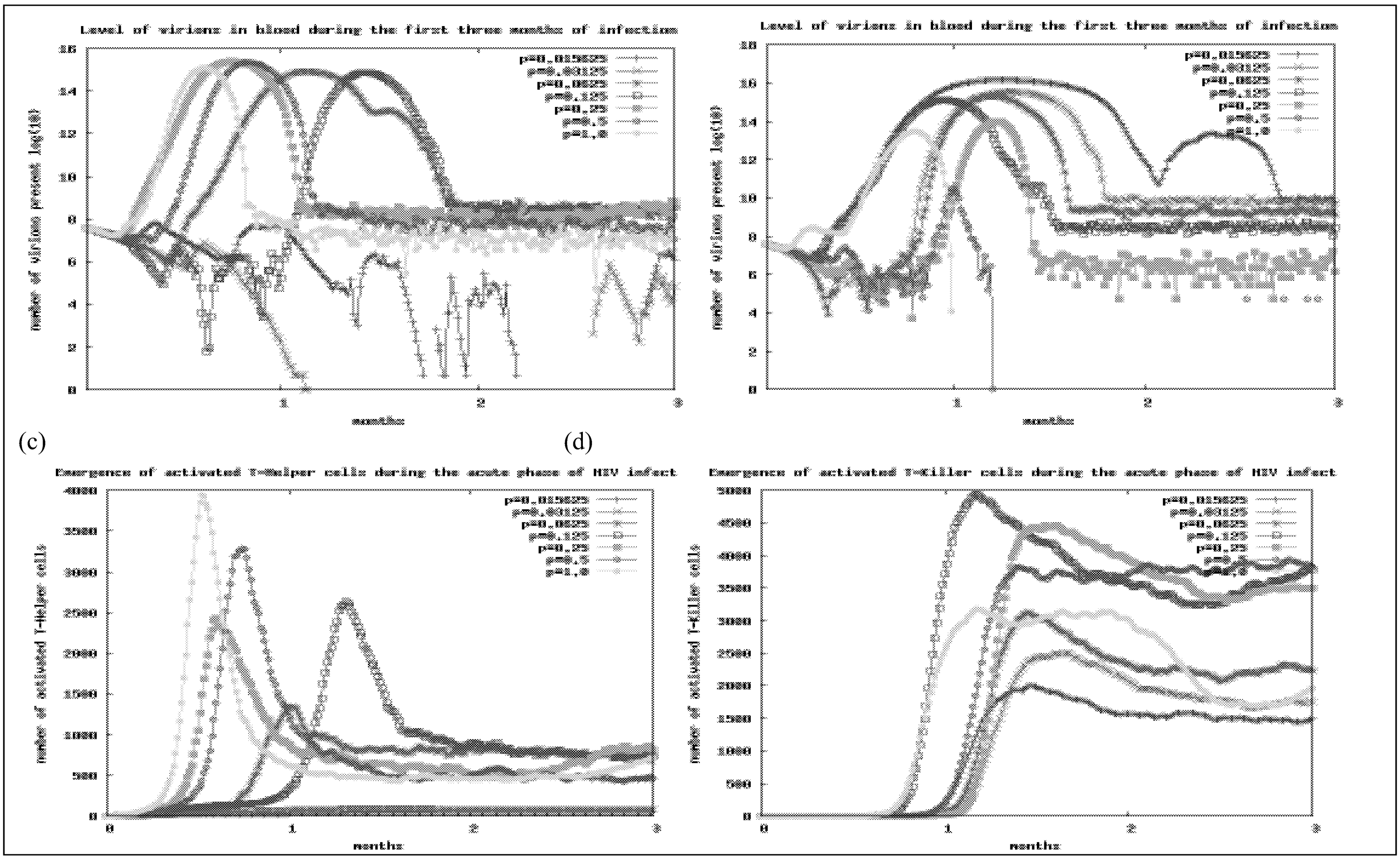


Fig 3: Graphs of viral dynamics as the ability to mount a humoral or cellular response is varied: (a) viral load per mL as the ability to mount a humoral response is varied: (b) viral load per mL as the ability to mount cellular response is varied(c) number of activated T Helper cells as ability to mount a humoral response is varied:(d) number of activated T Killer cells as ability to mount a cellular response is varied.

Before examining the relative importance of the humoral versus the cellular response with regard to infection with HIV, another result that emerged from these experiments deserves further discussion. Fig 3a depicts the level of viral load over the first three months of infection when the ability of the immune system to mount a humoral response is varied. What is of interest here is what happens when the probability of successfully presenting by means of the MHC2 complex is at the two lowest values. In both these cases the model fails to recreate the dynamics associated with the acute phase of HIV infection as described previously. The reason for this can be explained when one examines Fig. 3c in conjunction with Fig 3a. Fig3c depicts the number of activated T Helper cells as they emerge over the acute phase of infection; it shows that for the two lowest probabilities the number of activated T Helper cells is miniscule. As noted previously although HIV can infect various types of cells including macrophages and dendritic cells, its primary target is activated T Helper cells. The relative absence of activated T Helper cells causes a lack of target cells that HIV can infect and subsequently replicate within. As a result the level of HIV remains relatively low for much of the acute phase and does not reach significant levels until approximately three months after initial infection. It is important to note that this situation would not occur in reality where a pool of activated T Helper cells will always exist due to the continued exposure to foreign antigens whether they are parasites, viral infections or relatively harmless bacteria (Benjamini *et al* (2000), Leffell *et al* (1997)).

Fig. 3c and Fig. 3d depict what one would expect i.e. a significant level of variation in the level of activated T Killer cells and T Helper cells as the ability of the immune system to mount a successful cellular or humoral response respectively is varied. The relative importance of the humoral versus the cellular response can be seen when one compares Fig. 3a and Fig. 3b. Fig. 3a depicts the viral load as the ability of the immune system to mount the humoral arm of the immune response is varied, while Fig. 3b depicts the corresponding result with regard to the cellular response. Looking at Fig. 3b it can be seen that as the ability to mount a successful cellular response is varied there is significant variance in the level of viral load ($\approx 0 \leftrightarrow \approx 10^{11}$) at the end of the three-month period. In comparison Fig. 3a shows that as the ability to mount an effective humoral response is varied there is significantly less variation found in the level of viral load observed at the end of the acute phase. This is particularly true when the two most limiting cases discussed previously are ignored and doing so leaves the viral load in this case varying between approximately $10^7 \leftrightarrow 10^9$. This suggests that the ability to mount an effective cellular response plays a far more significant role than the ability to mount an effective humoral response in explaining individual variance observed in HIV infection. The results presented here support the hypotheses put forward by Letvin and Walker (2003), in which they postulate that the extent of the cellular response is the key factor in the early control of HIV replication.

DISCUSSION

The computer simulations of the dynamics of HIV and the immune system during the acute stage of infection carried out here have been successful in recreating the viral dynamics associated with this stage of the disease. In addition they have shown that the ability to mount a cellular response appears far more significant than the ability to mount a humoral response with regards to explaining the variation observed in different individuals infected with HIV. Limitations of the model include the very small number of cells we are currently able to model, and the fact that by using only 12 bits to represent the various receptors and molecules, the potential repertoire is far below what is found in reality. With regard to future work, parallel implementation of the model is currently underway. Parallel implementation of the model is desirable not only to model a more realistic number of cells and potential repertoire, (using parallel techniques we are able to use 28 bits as opposed to 12), but also to implement a compartmental model consisting of lymph nodes and peripheral blood, which will allow investigation of the impact, factors such as circulation patterns and initial location of viral entry may have on disease progression.

ACKNOWLEDGMENTS

Ashley Callaghan would like to acknowledge funding received from the Irish Research Council for Science, Engineering and Technology (IRCSET).

REFERENCES

J. Bailey, J.N Blankson, M. Wind-Rotolo and R.F Siliciano. 2004."Mechanisms of HIV-1 escape from immune responses and antiretroviral drugs." *Current Opinion in Immunology*, 16, 470-476.

E. Benjamini, R. Coico and G. Sunshine. 2000. *IMMUNOLOGY A Short Course,* Wiley & Sons, New York.

M. Bernaschi and F. Castiglione. 2001."Design and Implementation of an immune system simulator". *Computers in Biology and Medicine*, 31, 303-331.

J. Burns and H.J. Ruskin. 2003."A model of Immune Suppression and Repertoire Evolution**"**. in P Sloot and Y Gorbachev, eds, *Computational Science* - ICCS 2003, Vol. 2660 of *Lecture Notes in Computer Science*, Springer-Verlag, Berlin Heidelburg, 75–85.

J. Burns and H.J. Ruskin. 2004a. "Network topology in immune system shape space." in P Sloot and Y Gorbachev, eds, *Computational Science* - ICCS 2004, Vol. 3038 of *Lecture Notes in Computer Science*, Springer-Verlag, Berlin Heidelberg, 1094–1101.

J. Burns and H.J. Ruskin. 2004b. **"**Diversity emergence and dynamics during primary immune response: a shape space, physical space model**."** *Theory in Biosciences*, 123, 181-193.

F. Celada and P. Seiden. 1992. "A computer model of cellular interaction in the immune system." *Immunology Today*, 13, 56-62

J.M. Coffin. 1995. "HIV population dynamics in vivo: implications for genetic variation, pathogenesis, and therapy." *Science*, 267, 483-489.

C.A. Derdeyn and G. Silvestri. 2005. "Viral and host factors in the pathogenesis of HIV infection." *Current Opinion in Immunology*, 17, 366-373.

A.S Fauci. 1988. "The human immunodeficiency virus: infectivity and mechanisms of pathogenesis." *Science,* 239, 617–622.

W.C Greene and B.M Peterlin. 2002. "Charting HIV's remarkable voyage through the cell: Basic science as a passport to future therapy." *Science*, 239,617-622.

U. Hershberg, Y. Louzoun, H. Atlan and S. Solomon. 2001. "HIV time hierarchy: Winning the war while, loosing all the battles." *Physica A*, 289, 178-190.

M. Leffel, A.D. Donnenberg and N.R. Rose. 1997. *Handbook of HUMAN IMMUNOLOGY*, CRC Press, Boca Raton, Florida.

N.L Letvin and B.D Walker. 2003. "Immunopathogenesis and immunotherapy in AIDS virus infections". *Nature Medicine*, Vol. 9, No. 7, 861-866.

R. Mannion, R.B. Pandey and H.J. Ruskin. 2000. "Effect of cellular mobility on immune response." *Physica A*, 283, 447-450.

R. B. Pandey. 1991. "On a discrete approach to cell dynamics in context to immune response against HIV." *J. Phys. I France*, 1709-1713.

A.S. Perelson and G.F Oster. 1979. "Theoretical Studies of Clonal Selection: Minimal Antibody Repertoire Size and Reliability of Self-Non-Self Discrimination." *Journal Theoretical Biology*, 81 (4), 645-70.

A.S Perelson, A.U. Neumann, M. Markowitz, J.M. Leonard and D.D. Ho. 1996. "HIV-1 dynamics *in vivo*: virion clearance rate, infected cell life-span, and viral generation time." *Science*, 271, 1582-1586.

A.S Perelson and P.W Nelson. 1999. "Mathematical analysis of HIV-1 dynamics *in vivo*." *SIAM Rev.,* 41, 3-44.

H.J. Ruskin, R.B. Pandey and Y. Liu. 2002. "Viral load and stochastic mutation in a Monte Carlo simulation of HIV" *Physica A*, 311, 213-220.

M. Stevenson. 2003. "HIV-1 pathogenesis." *Nature Medicine*, Vol. 9, No. 7, 853-860.

R.M. Zorzenon dos Santos and S. Coutinho. 2001. On the Dynamics of the Evolution of HIV Infection, *Phys. Rev. Lett.,* 87, 31-34

SIMULATION OF ATTENTIONAL NETWORKS IN THE BRAIN – AN AGENT BASED APPROACH

Terje Kristensen and Jørgen Johansen

Department of Computer Engineering, Bergen University College,
N-5020 Bergen, Norway
Emails: tkr@hib.no; jorgen-j@broadpark.no

KEYWORDS: attention, agents, Multi-Agent Systems, Bergen Child Study, simulation

ABSTRACT

According to the Attention Network Model, three areas of the human brain are activated during attentional tasks. An Attention Network Test (ANT) has been developed to measure the performance of each of the networks. Based on the attention network theory and test results of the ANT, *ANTon*, a system for simulation the attentional process has been developed.

ANTon is implemented by a Multi-Agent System (MAS) with one agent for each of the attentional networks. Tasks are assigned to the agents according to the attention network theory such that the architecture of the MAS is comparable to the human brain.

1. INTRODUCTION

Bergen Child Study is a project related to children's mental health and development. It was started in 2002 and included examination and inquiry of children in 2nd to 4th grade of schools in Bergen, Norway.

The Bergen Child Study makes the necessary background for the development of the *ANTon* system – a framework for simulating the attentional process using software agents. One of the system's main objectives was to provide a configurable framework supporting the attention process of the brain, with abilities to produce results similar to human test subjects.

Attention is a complex cognitive function, dependent on interacting neural systems in the brain. It involves the process of selecting and holding information, and can be viewed as either a limited resource of information processing or some kind of filter, keeping unwanted information out, and letting wanted information through (Posner, Petersen 1990, Posner, Raichle 1994).

Attention can be compared to a spotlight that lights up a funnel shaped area, where the light is brightest in the center and fading as the distance from the center increases (Bernstein et.al 2000). The most attended object is located in the center, but one will easily register changes in the scene near the attended area. As the distance between the change and the attention becomes weaker, it is impossible to register any change at all

Attention can be overt or covert. Overt attention involves directing the sensory systems towards something, like moving your eyes to check the clock on a wall. Covert attention is used when you close your eyes and visualize some kind of scene. Voluntariness is another factor describing attention. It is goal-driven and is the attention purposely directed toward information of interest.

2. ATTENTION

In the process of obtaining a percept, registering it and producing a reaction to it, different areas in the brain are activated. According to the *attention network model*, there are three main areas, organized as three different biological neural networks, active during attentional tasks: the *alerting network* detecting a percept, the *orienting network* selecting and filtering input information and the *executive control network* identifying the percept and producing a reaction to it (Berger, Posner 2000).

2.1 Visual Attention

Visual attention is a stimulus-response process that occurs when a percept is received through the eyes. The percept is identified, and a response is produced. Visual attention can be spatially based or object based (Chokshi et.al. 2004).

Spatial attention is attention directed towards a limited area – attention fixed to certain positions. One focuses on detecting visual alterations in the given space. An example of spatial attention can be to monitor a doorway to see if someone is coming or going. Object based attention is attention connected to the properties of a given object, independent of position. The properties can be color, shape, etc..

According to the Attention Network Model (Posner, Petersen 1990), visual attention is associated with three operations: *alerting*, *orienting* and *executive functions*, located in different areas of the brain.

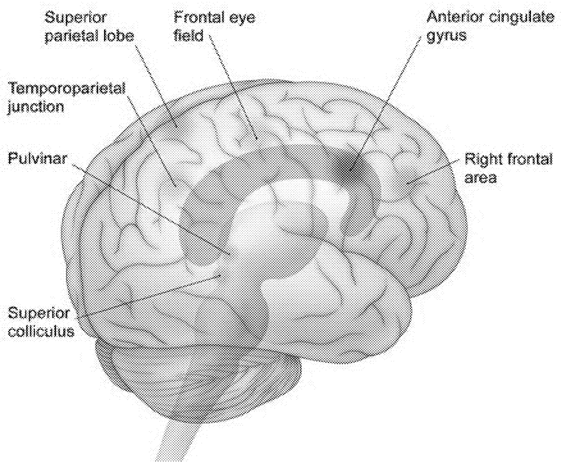


Fig.2.1 Attentional networks in the brain.

The alerting operations are responsible for maintaining a state of awareness, which decreases the reaction time. The right frontal lobe and the right parietal lobe in the brain are active during alerting operations as shown in figure 2.1.

Orienting operations involves perceiving sensory signals of visual sensory input and directing the attention towards these signals. The orienting functions are located in the parietal lobe and the ocular motor system of the brain

The executive operations includes goal directed behavior, goal- and error detection, conflict management and execution of automatic responses. During these operations, activity increases in the midline frontal areas. It is believed that basal ganglia control the connections between the executive functions and the other attentional functions. Basal ganglia are a collection of nuclei responsible for movement control.

2.2 Attention Network Test

The Attention Network Test (ANT) is a very simple test based on *Posner's cue-target test*. The ANT can be used to compute the activity in each neural network based on a set of measurements of the reaction time, manipulated by cues and different levels of interference. The test has been developed at the *Sackler Institute for Developmental Psychobiology* (Fan et.al 2002).

When taking the test, the subject (adult, child or primate) first stares at a computer screen with a centered cross. After some time (400-1600ms) a *cue* is presented for about 100ms, followed by a target phase where a number of arrows appear. The test objective is to identify the direction of the center arrow by giving input to the system, for instance by clicking the left or right mouse button. This procedure is repeated so that it is possible to make reliable average values for each of the input states.

The cue phase has four variations. No cue, center cue, double cue or spatial cue either located above or below the cross.

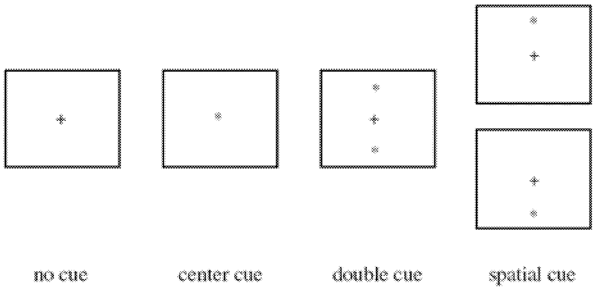


Fig. 2.2 The different variations of the cue phase.

In the target phase there are three variations: *neutral target*, with a center arrow flanked by plain lines, *congruent target* where all the arrows are pointing in the same direction and *incongruent target* where the flanking arrows are pointing in the opposite direction of the center arrow.

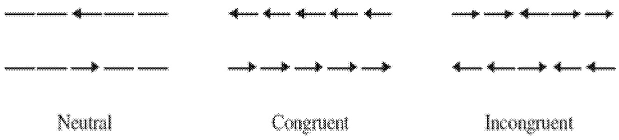


Fig. 2.3 Different types of cues.

Figure 2.4 shows the different phases of the test in chronological order.

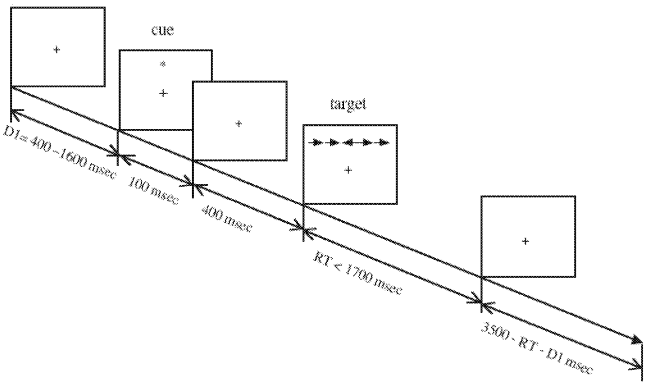


Fig. 2.4 ***The chronological phases of the ANT.***

2.3 Computation of times

The effect of the alerting network is computed by the difference between the average reaction time of the double cue instances, RT_{dbl} and the reaction time of the no-cue instances, RT_{no}. The average time of the alerting operation was 47ms with a standard error of 18ms in the Bergen Child Study.

The time consumption of the alerting network is given by:

$$t = RT_{no} - RT_{dbl} \tag{2.1}$$

The effect of the orienting network is computed by the difference between the average reaction time given a spatial cue, $RT_{spatial}$ and the average reaction time, given a center cue, R_{center}. The orienting state had an average reaction time of 51 ms with a standard error of 21 ms.

The time consumption of the orienting network is:

$$t = RT_{spatial} - RT_{center} \tag{2.2}$$

The time consumption of the executive control network is computed by the difference between the average of all congruent targets summed over all types of cues, RT_{incon}, and the average of all incongruent targets, RT_{con}. The executive control network has an average time consumption of 84 ms with a standard error of 25 ms.

The time consumption of the executive control network is then:

$$t = RT_{incon} - RT_{con} \tag{2.3}$$

While there is a high correlation between the total reaction times in the test, the computations are less reliable. The alerting computation is least reliable, while the computation of the executive control effect is most reliable. The reason is probably that the alerting- and orienting operations are dependent of the cue, and the executive control operations can be measured directly from the reaction.

Based on test results of 40 subjects it is believed that there is no correlation between the three biological neural networks. This implies independence between the three attentional networks, although there is an exception for the instances with no cue, giving a smaller extent of alerting. The total reaction time then increases, but the error rate decreases. Thus, it is assumed that less alerting gives more time to the executive functions.

2.4 The Attentional Process

When the visual system receives a stimulus, the visual cortex is activated. In turn, this activation increases activity of the alerting network. If the activity in the alerting network persists, it activates the orienting network and prepares it for the incoming stimuli. If the stimulus carries spatial information, an increase of the neural activity in the corresponding sub-area of the orienting network will occur.

The orienting network is connected to the object pathway, thus filtering the information. The filtering process is made by excitation of the neurons from the area in focus and inhibition of the neurons from the other areas. The filtered information in the end consists of a number of possible outputs. The executive control network selects the most suitable response.

As shown on figure 2.4, a perception is received through the primary visual cortex. The attentional process is started, activating the three attentional neural networks. There are interactions between the three networks and the *object pathway*, which is the component that is able to recognize objects. The object pathway component distinguishes between cues and arrows and directs the process flow according to the input. When an arrow is recognized, the executive control component selects a response from the set of possible responses. Instructions on how to execute the response are then returned.

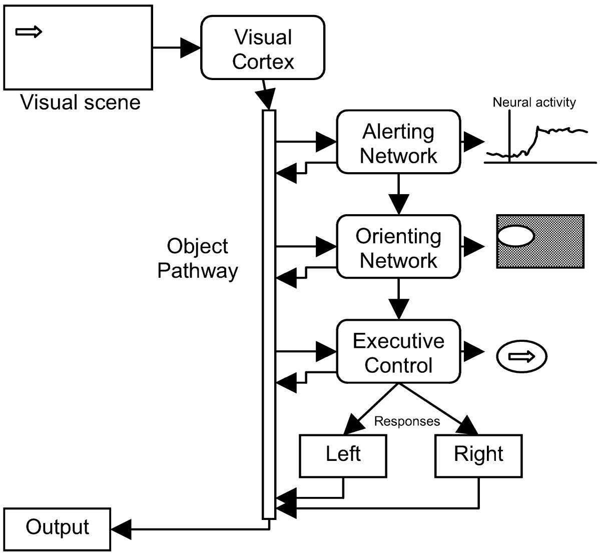


Fig. 2.4 The attentional process.

3. WHY USE AGENTS?

The goal of the ANTon system, described in this paper, is to support the attentional process outlined in section 2.4. The brain works as a highly parallel system to solve a wide range of tasks. Neuro-imaging has given many answers to which components are active and are cooperating during certain tasks

Studies have shown that the three attentional networks operate independently. This is hard to simulate using a sequential model, as the only way for interaction to happen between objects is by method invocation, and thus the invoking object gives up the ability to do processing, until the object is called upon or the currently active object completes its task. Parallel processing is thus an important feature in order for the simulation process to become successful.

Some biological neural networks are autonomous to a certain degree: when a double cue is presented, the focus has to change. If the test subject does not voluntarily select one of the cues or wanders between them, the focus changes involuntarily. This involuntary focus change is controlled by the orienting network, which indicates autonomy. However, it is important that the performance of the orienting network is good enough to support the neural networks and functions that depend on it.

Reactivity is one of the most important features of a biological neural network. A response is produced from a set of inputs (stimuli) very quickly. The brain is composed of a large number of neurons ordered in many neural networks, and every neuron is a stimulus-response unit that ensures reactive abilities throughout the brain.

When a task is performed many areas of the brain are active, as for instance the alerting-, orienting- and executive control functions during the attentional process. Connections and interaction between brain components are necessary to solve a problem. This social feature is also an ability that is shared with a software agent.

Agent technology provides excellent methods for dividing problems into sub-problems and building component based software. Each agent operates autonomously, and the sub-problems can be solved in a very structured way. The agent development framework makes sure the agents can communicate and cooperate, and the sub-problems can conveniently be put together to form a solution. The functions of each network can be implemented by using an agent, and the interaction between them can be modelled using predefined agent communication methods.

Most of the details on how the biological neural networks operate are still unknown. The attentional function is therefore hard to simulate. Thus, a multi-agent approach is useful for abstracting from the details and still providing a consistent interaction model or an external agent model. The implementation of the agents can be changed easily, and each agent could be implemented as an artificial neural network, thus simulating brain behavior close to its own architecture.

4. THE AGENT MODEL

The agent model was implemented in Java, using the JADE (Java Agent Development, http://jade.tilab.com) framework. JADE supports the FIPA (http://fipa.org) standards. The ANTon system has been constructed according to these standards.

The AAII analysis and design methodology (Kinny, Georgeff, Rao 1996) was used and four roles were identified in the domain: alerting, orienting, executive control and controller.

The alerting role is responsible for the level of alertness in the system. This will be achieved by monitoring for cues in the visual scene and to inform the other agents about the level of alertness. The alerting component provides an *alerting network service*.

The orienting role is responsible for controlling and changing the point of focus towards the areas of interests, when for instance the cue- and target are displayed. This will also be achieved by monitoring the visual scene. When the area of interest is in focus, its coordinates will be sent to the executive control component. The orienting component provides an *orienting network service*.

The executive control role is responsible for identifying objects located in the area of focus and to produce a response to these objects if required. When coordinates are received from the orienting component, the executive control component request the items displayed in the area around the coordinates. If a target is displayed, the executive control component produces a response which is sent to the controller. The executive control component provides an *executive control network service*.

The controller role is responsible for issuing test runs, providing information about the visual scene and collecting results after the run is completed. The network components are all reactive systems, and a run can be started simply by displaying a cue. This will trigger the alerting component and the orienting component. The executive control component will receive the point in focus and request the object displayed. The cue-object is returned, and the executive control component knows not to respond to a cue according to the instructions of the ANT. A target is displayed and again the alerting- and orienting component is reacting to it.

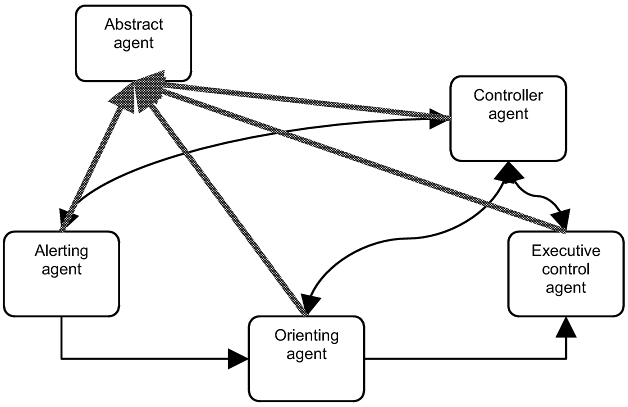


Fig. 3.1 The external agent model.

When the executive control component requests the object, the target is returned and a response is produced. When the controller receives the response, it starts to collect the information about the run, including the reaction times of each component. The controller component provides the *controller service.* Figure 3.1 shows the external agent model. The blue arrows describe the inheritance whereas the black arrows indicate interaction. The abstract agent defines an interface for the agents in the system and implements methods that are common for all the agents, including messaging and configuration management.

4.1 The Attention Network Agents

An agent influences the environment it works in by performing actions that can change the environment state. The *agent function* takes an environment state, or more precisely a run, R^E as input and produces an action.

$$Ag : R^E \rightarrow Ac \qquad (4.1)$$

An *agent program* implements the agent function. The function covers the whole decision-making procedure and its implementation determines how the agent will perform.

The agent function can be implemented in many ways. In some cases it is sufficient to list all the relevant environment states and a corresponding action for each state, but such an implementation hardly makes the agent intelligent. Besides, such an approach is impossible if the environment has a high- or infinite number of states.

4.1.1 The Alerting Agent

This agent controls the level of alertness throughout the system. The alerting level gets a boost from a cue and is weakened during idling. The alerting level is constantly sent to the orienting agent where the alerting level determines the effectiveness of the orienting operation.

The alerting agent constantly reviews the level of alertness and the agent function is defined as:

$$Ag_{\square}(e) = \begin{cases} \textit{increase alerting} & \textit{if cue received} \\ \textit{decrease alerting} & \textit{otherwise} \end{cases} \qquad (4.2)$$

4.1.2 The Orienting Agent

This agent is responsible for directing attention towards an area with information of a certain interest. It thus has the responsibility of filtering and selecting information. When there is a change in the view, the orienting agent shifts the focus to the area where the change occurred. The agent operates with a current point of focus and a target point and constantly seeks to coincide them.

The changes in the view are made either by a cue or a target, represented as incoming messages to the agent. Each message has a position and the target point of the orienting agent is set to this position. When the focus reaches the target, a message with information about the area in focus is sent to the executive control agent.

The agent function of the orienting agent can be described by the environment state e as:

$$Ag_{\square}(e) = \begin{cases} \textit{approach target} & \textit{if not on target} \\ \textit{do nothing} & \textit{otherwise} \end{cases} \qquad (4.3)$$

4. 1.3 The Executive Control Agent

When the orienting agent reaches its target, the executive control agent receives a message with the position of the area in focus. It then tries to identify the objects in this area. If a target object is identified, the agent tries to determine the direction of the center arrow and a response is produced. The time needed to identify the target is dependent on distracting elements in the view, such as the flanker arrows. Thus an incongruent target should be identified slower than a congruent- or neutral target. The chance of error also increases if there are many distractions in the view near the attended area.

$$Ag_c(e) = \begin{cases} \textit{request items} & \textit{if focus point received} \\ \textit{analyze item} & \textit{if item received} \\ \textit{produce response} & \textit{if target is identified} \end{cases} \quad (4.4)$$

4.2 Agent Communication

A message between agents is defined in a certain Agent Communication Language, ACL. FIPA is the default ACL of JADE. An ACL message has only one mandatory property, *performative*, identifying the *communication type* of the message. The use of performatives is closely connected to *speech-act theory*, where communication is regarded as actions (Wooldridge, 2002) and thus has the ability to make physical changes to the environment. Three conditions are required for a successful speech-act to be identified: first, the performative has to trigger a conventional procedure accepted by all participants of the conversation, second the procedure must be executed correctly and third the act must be sincere. Five classes of performatives are defined: inform, request, promise, thanks and declaration.

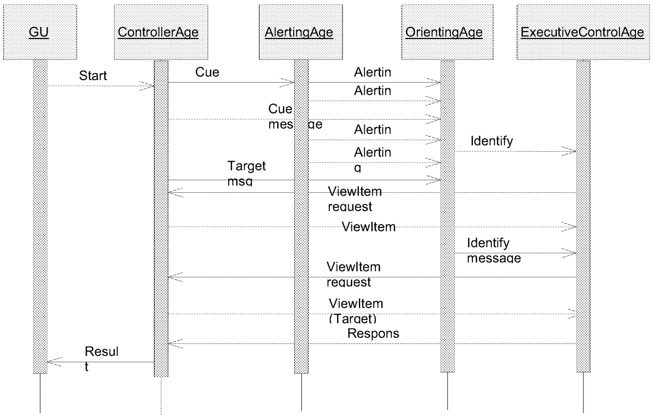


Fig. 4.1 Communication during a run.

The performatives of FIPA are based on the classes listed above. Only two performatives are used in the ANTon system: INFORM and REQUEST. These are the fundamental performatives of FIPA and also the most general. In addition there are performatives for queries, proposals, subscriptions, etc. The collection of performatives fully supports every protocol specified by FIPA, such as auctions with a number of courses, task distribution and negotiations.

An ACL message should also contain information about the sender and the receiver. Message content can also be set and information about how the content should be handled.

FIPA has also specifications for content language, interaction- and conversation protocols and ontologies. An ontology defines a set of concepts within a domain. If an ontology is selected for a message, communication protocol or conversation, all the participants have to know the ontology. The ontology thus works as an extended namespace, but in addition to strings, an ontology can also define object structures.

4.3 Changing the Implementation

All the agents are acting individually in the multi-agent system and as long as the necessary services for the ANT process is available for each test run, individual agents can be replaced, added and even removed and the process will still be able to complete. As long as each agent implement the interface that is defined for the task the agent is responsible for, the attention process will be able to complete and results will be returned. This way is excellent for testing different methods for producing results with a minimum of changes to the implementation necessary.

5. JADE AND BIOLOGICAL SIMULATION

The JADE framework provides an *agent platform*, an environment the agents can work in. Each agent has to be instantiated in an *agent container*. An agent container is simply a proxy class, providing access to the agents instantiated in it. The platform has a *main container* with an Agent Management System-agent providing a naming service for the other agents on the platform and a *Directory Facilitator-agent*, with a *Yellow Pages-service*, helping the agents to get access to services provided by other agents. JADE is implemented by the FIPA standards, and it is thus possible for JADE agents to interact with agents from any other agent framework implementing these standards.

The framework includes many features that are specific for the business- and economic domain, also supported by the FIPA standards. The collection of performatives further indicates the connection to the business domain, as most of the performatives deal with negotiation, cooperation and service exchange.

When simulating biological processes using agent technology, the close connection to business-specific standards that is implemented by JADE, reduces the possibilities to create optimal models.

A multi-agent based framework specifically designed for simulations of biological systems would be beneficial. In such an agent framework the concepts could be redefined to better fit the biological domain and interaction protocols could be defined to support a greater resemblance with the information flow in these systems.

Further, a layered model could be created to support implementations in the range from simple high level computations to lower level artificial neural networks. The layered model stating the level of implementation detail could be the basis of creating an agent typology for biological agents similar to the many existing agent typologies classifying agents in the business domain.

The agent types should be created, reflecting the biological components of functions they represent, thus creating a bond between the agents and the biological domain, similar to the bonding between existing standards and the business domain.

6. DISCUSSION

One of the main goals of the ANTon system was to produce results accordingly to results made by human test subjects. So far, such results have not been produced.

Whereas the external agent model, specifying the interactions between the agents, are based on existing theories about the attentional process and the interactions between the attentional networks, little is known about the internal structure of the neural networks of the brain. The key to produce better results thus lie in the internal agent model.

The internal models of the agents in the ANTon system are very simple, and it is difficult to compute the time spent doing the tasks. Because each agent's task is so small, it is impossible to use the amount of work as a measure for time consumption. The alerting agent can be used as an example: its only responsibility is to manage the level of alertness throughout the system and respond to alerting level requests from the other agents. When a stimulus is received the alerting level changes, and a numeric value in the internal agent model is changed. How can such work be simulated so that it justifies a time consumption of for instance about 80 ms?

In the main implementation, the reaction times for each network are drawn from a Gaussian distribution with a mean value and standard deviance extracted from the test results of the *Bergen Child Study*. However, alternative implementations of the alerting agent and the orienting agent have been made using a *time block* approach. This approach combines the amount of work and the time spent in a better way. Instead of changing the values in the internal agent models using one operation, functions for iteratively approaching a target value is introduced. For each iteration a small amount of time is added to the total time consumption which is returned as the network's reaction time. A variant of the time block approach, where delays in the internal processing of each agent is inserted, has also been considered. This *real time* approach ensures that the system uses the same amount of time as human test subjects.

The ANTon system can also be used to simulate other attentional tasks with some extensions. A memory component could be added, providing the executive control component with instructions on when responses should be produced and how to respond to certain input. Such an extension would make the system more suitable for more general tasks.

7. CONCLUSION

One of the most important features by using agent technology is the flexibility that agents provide. Whereas many details are unknown about the internal structure of each component of the attentional process, an agent based approach can be used to test the effects of changing the internal models of the agents.

The existing standards of agent technology are closely related to the business domain. New concepts should be developed to better describe the operation of biological systems, the workflow, the information flow between different components and the methods of interaction.

Agents have many features that easily can be adopted to describe biological systems. It is therefore natural to consider use of software agents to simulate biological components. A better understanding of biological systems can be achieved by using such a software framework. More intelligent systems may then also be developed..

References

Berger A, Posner MI. Pathologies of Brain Attentional Networks. Neurosci Biobehavior, Rev 24, 2000.

Bernstein, Penner, Clarke-Stewart & Roy. Psychology 5th edition. Houghton Miffin Company, 2000.

Chokshi K, Panchev C, Wermter S, Taylor JG. Knowing What and Where: A Computational Model of Visual Attention. Proceedings of IEEE International Joint Conference on Neural Networks, 2004.

Fan J et. al. Testing the Efficiency and Independence of Attentional Networks. Journal of Cognitive Neuroscience, page 340-347, 2002.

Kinny D, Georgeff M, Rao A. A Methodology and Modelling Technique for Systems of BDI Agents. 7th European Workshop on Modelling Autonomous Agents in a Multi-Agent World, 1996.

Posner MI, Petersen SE. The attention system of the human brain. Annual Review of Neuroscience 13, 1990.

Posner MI, Raichle ME. Images of Mind. Freeman 1994

Wooldridge M. An Introduction to Multi-Agent Systems. Wiley 2002.

BIOGRAPHY

TERJE KRISTENSEN is an associate professor in informatics and mathematics at Bergen University College, Norway. He has published lots of international papers and many books. He has also been session chair and member of many international program committees. His special interests are neural networks, machine learning, multi-agent systems and simulation. He is also director of the company Pattern Solutions which is developing pattern recognition applications (http://www.patternsolutions.no)..

JØRGEN JOHANSEN has a master degree in software engineering from Bergen University College. His research interests include Multi-Agent Systems, brain processing and software engineering.

AGENT
BASED
NEGOTIATION

Simulation of an Agent-based MarketPlace

Maria João Viamonte, Isabel Praça, Carlos Ramos, Zita Vale

GECAD –Grupo de Investigação em Engenharia do Conhecimento e Apoio à Decisão
Instituto Superior de Engenharia / Instituto Politécnico do Porto
Rua Dr. António Bernardino de Almeida, 4200-072 Porto, Portugal
Fax: +351-228321159, Phone: +351-228340500

Emails: mjv@isep.ipp.pt; isp@dei.isep.ipp.pt; csr@dei.isep.ipp.pt; zav@dee.isep.ipp.pt

KEYWORDS: Intelligent Agents, Simulation, Negotiation, Decision-Making.

ABSTRACT

This paper presents a Multi-Agent Market simulator designed for analysing agent market strategies based on a complete understanding of buyer and seller behaviours, preference models and pricing algorithms, considering user risk preferences and game theory for scenario analysis. The system includes agents that are capable of improving their performance with their own experience, by adapting to the market conditions.

1. INTRODUCTION

Unlike traditional tools, agent based simulation does not postulate a single decision maker with a single objective for the entire system. Rather, agents, representing the different independent entities in electronic markets, are allowed to establish their own objectives and decision rules. Moreover, as the simulation progresses, agents can adapt their strategies, based on the success or failure of previous efforts.

Some approaches to agent-based simulation applications for competitive electronic markets are more targeted or limited than our proposal: some of them address only one negotiation type; do not consider behaviour dependent dynamic strategies, or expected future reactions; other, although considering behaviour dependent dynamic strategies, frequently assume that agents have complete information about market, such as the distribution of buyer preferences or its competitor's prices (Dasgupta and Das, 2000), (Sim and Choi, 2003), (North et al., 2002), (Monclar and Quatrain, 2001).

We present a multi-agent market simulator designed for analysing agent market strategies based on a complete understanding of buyer and seller behaviours, preference models and pricing algorithms, considering user risk preferences and game theory for scenario analysis. Each market participant has its own business objectives, and decision model. The results of the negotiations between agents are analysed by data mining algorithms in order to extract knowledge that give agents feedback to improve their strategies. The extracted knowledge will be used to set up probable scenarios, analysed by means of simulation and game theory decision criteria.

We intend to apply this platform to different market types, taking into account some previous work of our research group, where two different simulation platforms have already been developed, namely ISEM – Intelligent System for Electronic MarketPlaces (Viamonte et al., 2004), and MASCEM – Multi-Agent Simulator for Competitive Electricity Markets (Praça et al., 2005).

ISEM focuses specially on markets with finite time horizon. This simulator was recently selected as a worldwide case study in simulation of negotiation agents (Viamonte et al., 2006), while MASCEM focus specially on market mechanisms usually found in liberalized electricity markets and was selected as a worldwide case study of agents technology applied to markets (Praça et al., 2003).

Our proposal is a Market Simulator that will act as a kind of What-if tool, trying to analyse what may occur if some decision is taken. However, some additional intelligence need to be placed in the system, otherwise we will have a kind of combinatorial explosion, since many scenarios need to be analysed. Moreover, the Market Simulator will be used as the engine of a Market Participant (Seller or Client) in order to suggest him/her about the actions to have in the market.

Entities from real markets can use our tool to test several different negotiation mechanisms, different behaviours, strategies and risk preferences, and to analyse the future market evolution and other entities expected reactions.

2. MULTI-AGENT MODEL

Our Simulator facilitates agent meeting and matching, besides supporting the negotiation model. In order to have results and feedback to improve the negotiation models and consequently the behaviour of user agents, we simulate a series of negotiation periods, $D = \{1,2,...,n\}$, where each one is composed by a fixed interval of time $T = \{0,1,...,m\}$.

Furthermore, each agent has a deadline $D_{\max}^{Agt} \in D$ to achieve its business objectives. At a particular negotiation period, each agent has an objective that specifies its intention to buy or sell a particular good or service and on what conditions.

The available agents can establish their own objectives and decision rules. Moreover, they can adapt their strategies as the simulation progresses on the basis of previous effort's successes or failures. The simulator probes the conditions and the effects of market rules, by simulating the participant's strategic behaviour.

The simulator was developed based on "A Model for Developing a MarketPlace with Software Agents (MoDeMA)" (Viamonte 2004). Multi-agent model includes a market administrator, buyers, sellers, traders and a market operator. Figure 1 illustrates the Multi-Agent Model.

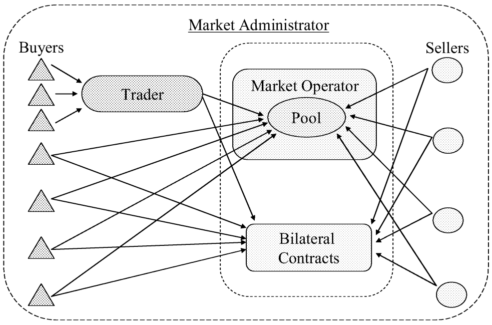


Fig. 1 MarketPlace Model

The market administrator agent has two main functions: coordinator and knowledge provider. On one hand it coordinates the simulated market and ensures that it functions correctly, according to market mechanisms and established rules. On the other hand, it plays the role of "power" agent, since it has access to market knowledge, which contains information about the organisational and operational rules of the market, as well as information about all different running agents, their capabilities and historical information. The market previsions and agent behaviour models are obtained through data mining algorithms, using data resulting from agent negotiations that support agents' market strategies (see details in section 5).

Since we intend to cover several negotiation mechanisms, our model also includes a market operator agent, responsible to support negotiations based on an auction mechanism.

Seller and buyer agents are the two key players in the market, so we devote special attention to them, particularly to their business objectives and strategies to reach them. In order to be competitive in today's economic markets, buyer and seller agents need not only to be efficient in their business field, but also to be able to quickly react and adapt to new environments as well as to interact with other available entities. The control architecture adopted for the design of those agents meet these requirements, having a similar structure but with a kind of symmetrical behaviour (due to their antagonistic business objectives).

3. NEGOTIATION MECHANISMS

As a decision support tool, our simulator includes several types of negotiation mechanisms to let the user test them and learn the best way to negotiate in each one. So, we include bilateral contracts and a Pool, centralized mechanism based on an auction, and regulated by a market operator. Both types of negotiation may exist at the same time: Mixed Market. These implies each agent must decide whether to, and how to, participate in each market type.

Let $Agtb$ denote the buyer agent, $Agts$ the seller agent and let $[Pi_{\min}, Pi_{\max}]$ denote the range of values for price that are acceptable for agents.

A seller agent has the range $[Psi_{\min}, Psi_{\max}]$, which denotes the scale of values that are comprised of the minimum value that the seller is disposed to sell to the optimal value.

A buyer agent has the range $[Pbi_{\min}, Pbi_{\max}]$, which denotes the scale of values that are comprised of the optimal value to buy to the maximum value.

3.1 *Bilateral Contracts*

In bilateral contracting buyer agents are looking for sellers that can provide them the desired products at the best price. We adopt what is basically an alternating protocol (Faratin et al., 1998).

Negotiation starts when a buyer agent sends a request for proposal. In response, a seller agent analyses its own capabilities, current availability, and past experiences and formulates a proposal.

Sellers can formulate two kinds of proposals: a proposal for the product requested; or a proposal for a related product, according to the buyer preference model.

$PPg_{i\,Agts \to Agtb}^{DT}$ represents the proposal offered by the seller agent $Agts$ to the buyer agent $Agtb$ at time *T*, at the negotiation period *D* for a specific product.

The buyer agent evaluates the proposals received with an algorithm that calculates the utility for each one, U_{PPgi}^{Agtb}; if the value of U_{PPgi}^{Agtb} for $PPg_{i\,Agts \to Agtb}^{DT}$ at time T is greater than the value of the counter-proposal that the buyer agent will formulate for the next time *T*, in the same negotiation period *D*, then the buyer agent accepts the offer and negotiation ends successfully in an agreement; otherwise a counter-proposal $CPg_{i\,Agtb \to Agts}^{DT}$ is made by the buyer agent to the next time *T*.

The seller agent will accept a buyer counter-proposal if the value of U_{CPgi}^{Agts} is greater than the value of the counter-proposal that the seller agent will formulate for the next time *T*; otherwise the seller agent rejects the counter-proposal.

On the basis of the bilateral agreements made among market players and lessons learned from previous bid rounds, both agents revise their strategies for the next negotiation rounds and update their individual knowledge module.

3.2 *Pool*

In our simulator, agents also have the possibility of negotiating through a Pool, which is a centralized mechanism that functions according to an auction mechanism, and is regulated by a market operator. We have two different auction mechanisms: a double and a single uniform auction.

The process starts at the market operator, who sends a request for participation. The *call_for_participation* message triggers the negotiation process and is delivered to all agents in the simulated market. If the agent is interested, or capable, of participating in the Pool, it will formulate a bid and send it to the market operator, specifying for each requested parameter the value of its proposal.

The process of formulating bids, by buyer and seller agents, is related to agent strategies, addressed in detail in section 6. The market operator evaluates all the received bids, analyses them through the pool auction mechanism, defines the market price and accepted bids. Then a *reply_bid* message is sent to all pool participants, specifying the settled market price and if the bid was or not accepted and why.

3.3 *Mixed Markets*

The Mixed model combines features of Pools and Bilateral Contracts. In this model, a Pool isn't mandatory, and customers can either negotiate an agreement directly with sellers, at the pool market price or both. Agents must decide whether to try or not the Pool, whether to keep bilateral negotiations simultaneously with Pool negotiations or just after Pool results if bids were not accepted. For that agents use their past experiences, market knowledge and agents own negotiation strategies to support their decisions.

4. DATA MINING

The market previsions and agent behaviour models are obtained through data mining algorithms, using data resulting from agent negotiations that support agents' market strategies. In practice, usually, after a confidential negotiation period, the market administrator agent discloses information about past transactions and agents' characteristics (if possible); all agent interactions are logged at a transaction level of detail, which provide a rich source of business insight that can help to customise the business offerings to the needs of the individual buyers. With this functionality it is possible to discover sub-groups that behave independently and associations between products. For that, our market simulator uses clustering, classification and association operations.

To carry out the clustering operation a Two-Step clustering algorithm (Zhang et al., 1996) is used to target buyers with similar characteristics in the same agent group. Then, to obtain more relevant information that describes the consumption patterns of each cluster population, a rule-based modelling technique, using C5.0 classification algorithm, an evolution of C4.5 algorithm (Quinlan 1996), is used to analyse those clusters and to obtain descriptions based on a set of attributes, collected in the individual agents' knowledge module. These models are transferred to the market administrator agent and offer a set of market information, such as: preferred sellers; preferred marks; favourite products and reference prices, which support the process of agents' strategy implementation.

To discover associations between buyer details and purchases, data from multiple agent negotiations are manipulated to create "basket" records showing product purchases. This permits the observation of the behaviour of each buyer agent. This data is combined and manipulated by the "Apriori algorithm" (Agrawal et al., 1996), to discover associations between buyer details and purchases. The best association rules, those with a strong support and confidence, are extracted and transferred to the market administrator agent. With this kind of knowledge it is possible to provide insight into the sellers' agents about the profiles of buyer agents with certain purchase propensities, showing associations between products, prices, style, etc.

After these operations, to get confident data, agents can request the services provided by the market administrator agent, in order to support their strategic behaviour. Only players with more sophisticated behaviour will take advantage of this new knowledge; since the user can determine which seller agents have access to this facility. The user can also determine if the agents' information will be private or public; public information is available to market analysis with the data mining functionality. However the market can get knowledge about an agents' behaviour even if they are set as a private information agent. This situation occurs, by the simple fact of being on the market.

5. STRATEGIC BEHAVIOUR

5.1 *Time-dependent Strategies*

Agents use four *time-dependent strategies* to change their price during a negotiation period: *Determined*, *Anxious*, *Moderate* and *Gluttonous*, these strategies depending on both the point in time when the agent starts to modify the price and the amount it changes. In this work, we have also used the *time-dependent strategies*, based on the model proposed by S. Fatima (Fatima et al., 2004), to model different attitudes towards time, during a negotiation period. Although *time-dependent strategies* are simple to understand and implement (Morris et al., 2003), they are very important since they allow the simulation of important issues such as: emotional aspects and different risk behaviours. For example, an agent that gains utility, with the time, and has the incentive to reach a late agreement (within the remaining time until the end of a negotiation period) is considered a strong or patient player; an agent that loses utility with time and that tries to reach an early agreement is considered a weak or impatient player.

5.2 *Behaviour-dependent Strategies*

Agents use *behaviour-dependent strategies* to adjust parameters for the next negotiation period according to the results obtained in the previous ones. Buyers and seller agents develop their behaviour and strategies based on a combination of public information, available through requesting from market administrator services; and private information, available only to the specific agent at their individual knowledge module.

For Pool Negotiations we define two different behaviour-dependent strategies: one called *Composed Goal Directed* (CGD) and another called *Adapted Derivative Following* (ADF). The CGD strategy is based on two consecutive objectives, the first one is selling (or buying) all the available (or needed) units, and then increase the profit (reduce the payoff). The ADF strategy is based on the *Derivative Following* strategy proposed by Greenwald (Greenwald and Kephart, 1999). The ADF strategy adjusts its price by looking at the amount of revenue earned in the previous period as a result of the previous period's price change. If the last period's price change produced more revenue per good than the previous period, then the strategy makes a similar change in price. If the previous change produced less revenue per good, then the strategy makes an opposite price change.

For Bilateral Contracts Negotiations we also have several *behaviour-dependent strategies*. Buyer agents can use two complementary behaviour-dependent strategies: the *Modified Goal Directed for Buyers* (MGDB) and the *Fragmented Demand* (FD). The MGDB strategy is an adaptation of CGD for bilateral contracts. The FD strategy, adjusts the demand per day by attempting to reach the goal of buying its entire needs by the last day of the market, and not before, this strategy paces its purchases over the market, with the goal of buying all the units needed but with less costs. Seller agents can also choose from two different *behaviour-dependent strategies*: the *Modified Goal Directed for Sellers* (MGDS), that adjusts its price by attempting to reach the goal of selling the entire inventory by the last day of the market, by lowering prices when sales in the previous day are low and raising prices when the sales are high; and the *Derivative Following* (DF) strategy weighted by *Seller Satisfaction* (DFWS) or by the *Previewed Demand* for a specific product (DFWPD). The DFWS/PD is based on the ADF behaviour weighted by the referred issues. Seller agents can obtain these values through requesting for market administrator agent support.

5.3 ***Scenario Analysis Algorithm***

To obtain an efficient decision support, seller and buyer agents also have the capability of using the *Scenario Analysis Algorithm*. This algorithm provides a more complex support to develop and implement dynamic pricing strategies since each agent analyses and develops a strategic bid, for the next period, taking into account not only its previous results but also other *players* possible results and expected future reactions. It is particularly suitable for markets based on a Pool or for Mixed Markets, to support sellers and buyers decisions for proposing bids to the Pool and accepting or not a bilateral agreement. The algorithm is based on analysing several bids under different scenarios, constructing a matrix with the simulated results and applying a decision method to select the bid to propose.

Each agent uses historical information about market behaviour and about other agents' characteristics and behaviour, and information provided by the market administrator, by means of Data Mining techniques. With the information gathered agents can build a profile of other agents based on their expected proposed prices, limit prices, and capacities. With these profiles, and based on the agent own objectives and user risk preference, several scenarios, and the possible advantageous bids for each one, are defined.

We call a *play* to a pair bid-scenario. After defining all the scenarios and bids, market simulation is applied to build a matrix with the expected results for each play. The matrix analysis with the simulated plays' results is inspired by the Game Theory concepts for a pure-strategy two-player game, assuming each player seeks to minimize the maximum possible loss or maximize the minimum possible gain (Fudenberg and Tirole, 1991). Several decision methods are available, for example: a Seller trying to maximize the minimum possible gain may use the MaxiMin decision method; a Buyer trying to obtain the smallest maximum payoff may use the MiniMax decision method.

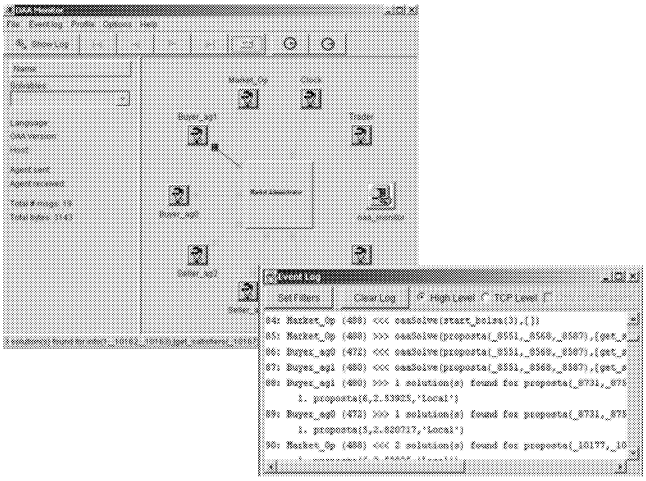
The analysis of each period's results will update the agent's market knowledge and the scenarios to study. After each negotiation period, instead of considering how other agents might increase, decrease, or maintain their bid, agents use knowledge rules that restrict modifications on the basis of other agents' expected behaviour. The knowledge rules update agents' bids in each scenario, but the number of scenarios remains the same. If at the end of a negotiation period the agent concludes-by analysing market results-that it incorrectly evaluated other agents' behaviour, it will fix other agents' profiles on the basis of the calculated deviation from real results and will also update the knowledge rules.

6. IMPLEMENTATION

A prototype was developed in Open Agent Architecture (OAA) and in Java. OAA (http://www.ai.sri.com/~oaa/) is a framework for integrating a community of heterogeneous software agents in a distributed environment. It is structured to minimize the effort involved in creating new agents, written in various languages and operating platforms; to encourage the reuse of existing agents; and to allow the creation of dynamic and flexible agent communities. The OAA's Interagent Communication Language is the interface and communication language shared by all agents, no matter which machine they are running on or which programming language they are programmed in. OAA is not a framework specifically devoted to develop simulations; some extensions were made to make it more suitable, such as the inclusion of a clock to introduce the time evolution mechanism of the simulation.

Each agent is implemented in Java, as a Java thread. The model can be distributed over a network of computers, which is a very important advantage to increase simulation runs for scenarios with a huge amount of agents.

Figure 2 illustrates the interface of the MarketPlace. With this interface the user can analyse the messages exchanged between all the players during a simulation run.



```
84: Market_Op (480) <<< oaaSolve(start_bolsa(3),[])
85: Market_Op (480) >>> oaaSolve(proposta(_8551,_8569,_8587),[get_s
86: Buyer_ag0 (472) <<< oaaSolve(proposta(_8551,_8569,_8587),[get_s
87: Buyer_ag1 (480) <<< oaaSolve(proposta(_8551,_8569,_8587),[get_s
88: Buyer_ag1 (480) >>> 1 solution(s) found for proposta(_8731,_873
     1. proposta(6,2.53925,'Local')
89: Buyer_ag0 (472) >>> 1 solution(s) found for proposta(_8731,_873
     1. proposta(5,2.820717,'Local')
90: Market_Op (480) <<< 2 solution(s) found for proposta(_10177,_10
```

```
3 solution(s) found for start(_10162,_10163),[get_satisfiers(_1016
```

Fig. 2 MarketPlace Interface

7. CONCLUSIONS

In the near future, agent market strategies will be a common competitive manoeuvre for Electronic Markets. Market participant's strategic behaviour is very significant in the context of competition. In addition, the availability of new market knowledge obtained with Data mining algorithms is vital for supporting marketing and sales. Also important is the development of agent-based tools that will help in understanding what kinds of electronic market strategies are appropriate. Very relevant is the availability of a Scenario Analysis Algorithm which is a promising algorithm to improve agents bidding process and counter-proposals definition. This analysis gives the agent not only decision support about the best bid to propose in a Pool but also makes possible the improvement of the negotiation mechanism for establishing bilateral contracts.

ACKNOWLEDGEMENTS

The authors would like to acknowledge FCT, FEDER, POCTI, POSI, POCI and POSC for their support to R&D Projects and GECAD Unit.

8. REFERENCES

Agrawal, R., Manilla, H., Srikantand, R., Toivonen, H. and Verkamo, I.: "Fast discovery of association rules". AAAI Press 1996.

Dasgupta, P. and Das, R.: "Dynamic Service Pricing for Brokers in a Multi-Agent Economy". Proceedings of the Third International Conference for Multi-Agent Systems (ICMAS), Boston, MA, 2000.

Faratin, P., Sierra, C. and Jennings, N.: "Negotiation Decision Functions for Autonomous Agents". Int. J. Robotics and Autonomous System, Vol.24 (3), pp. 159-182, 1998.

Fatima, S., Wooldridge, M. and Jennings, N.: "An agenda-based framework for multi-issue negotiation". Artificial In-telligence Journal vol. 152, nº1, pp.1-45, 2004.

Fudenberg, D. and Tirole, J.: "Game Theory". MIT Press, 1991.

Greenwald, A. and Kephart, J.: "Shopbots and Pricebots". Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence - IJCAI, Stockholm, 1999.

Monclar, F.R. and Quatrain, R.: "Simulation of Electricity Markets: A Multi-Agent Approach," Proceedings of the International Conference on Intelligent System Application to Power Systems - ISAP 01, pp. 207–212, 2001.

Morris, J., Greenwald, A. and Maes, P.: "Learning Curve: A Simulation-based Approach to Dynamic Pricing". Electronic Commerce Research: Special Issue on Aspects of Internet Agent-based E-Business Systems. Kluwer Academic Publishers, Vol. 3(3-4), pp. 245-276, 2003.

North M. et al.: "E-Laboratories: Agent-Based Modelling of Electricity Markets," Proc. American Power Conference, PennWell Corp., 2002.

Praça, I., Ramos, C., Vale, Z. and Cordeiro, M.: "MASCEM: A Multiagent System that Simulates Competitive Electricity Markets". IEEE Intelligent Systems, vol.18, nº 6, pp.54-60, Nov/Dec 2003.

Praça, I., Ramos, C., Vale, Z. and Cordeiro, M.: "Intelligent Agents for Negotiation and Game-based Decision Support in Electricity Markets", in International Journal of Engineering Intelligent Systems, Special Issue "Intelligent Systems Application to Power Systems", vol. 13, nº 2, pp. 147-154, 2005.

Quinlan, J.: "C4.5: Programs for Machine Learning". Morgan Kaufmann Publishers, San Francisco, USA, 1993.

Sim, K. M. and Choi, C. Y.: "Agents that React to Changing Market Situations" IEEE Transactions on Systems, Man, and Cybernetics – Part B: Cybernetics, vol. 33, nº2, 188-201, 2003.

Viamonte, M. J., Ramos, C., Rodrigues, F. and Cardoso, J. C.: "Simulating the Behaviour of Electronic MarketPlaces with an Agent-based Approach", Proceedings of the "International Conference on Web Intelligence" – IEEE/WIC 2004, ISBN 0-7695-1932-6, pp. 553-557, Beijing, China, pp. 553-557, 2004.

Viamonte, M. J.: "Mercados Electrónicos Baseados em Agentes – Uma Abordagem com Estratégias Dinâmicas e Orientada ao Conhecimento". PhD Thesis 2004.

Viamonte, M. J., Ramos, C., Rodrigues, F. and Cardoso, J. C.: "ISEM: A Multi-Agent Simulator For Testing Agent Market Strategies", IEEE Transactions on Systems, Man and Cybernetics – Part C: Special Issue on Game-theoretic Analysis and Stochastic Simulation of Negotiation Agents, Vol. 36(1), pp 107-113, Jan. 2006.

Zhang, T., Ramakrishnon, R. and Livny, M.: "BIRCH: An Efficient Data Clustering Method for Very Large Databases". Proceedings of the ACM SIGMOD Conference on Management of Data, p. 103-114, Montreal, Canada. 1996.

AN APPROACH OF AGENT-BASED DISTRIBUTED SIMULATION FOR SUPPLY CHAINS: NEGOTIATION PROTOCOLS BETWEEN COLLABORATIVE AGENTS

El Habib Nfaoui
Omar El Beqqali*
GRMS2I – FSDM/ Sidi Md Ben
AbdEllah University, B.P 1796
Fès-Atlas. Morocco

Yacine Ouzrout
Abdelaziz Bouras
PRISMa – University of Lumière
Lyon 2, 160 Bd of university, Bron
CEDEX 69676. Lyon France

firstname.lastname@univ-lyon2.fr
*obekkali@fsdmfes.ac.ma

KEYWORDS
Negotiation Protocols, Multi-Agent Systems, Supply Chain Management, Distributed and Proactive Simulation, Agent-UML, Modeling.

ABSTRACT

The modeling languages of multi-agent systems are recently developing, mainly AUML (Agent Unified Modeling Language). This paper presents modeling work based on the AUML language for distributed architecture of simulation and decision-making in the supply chain. The environment of supply chain is rich in negotiation protocols; we propose and model with AUML some exchange and negotiation protocols for agents within the supply chain context. We show through an example that AUML language could be used for specifying and modeling real-world agent-based applications.

INTRODUCTION

The economic and industrial communities worldwide are confronted with the increasing impact of competitive pressures resulting from the globalization of markets and supply chains (SC) for product fulfillment. More and more enterprises are being driven to pursue new forms of collaboration and partnership with their direct logistics counterparts. As a result, at a company level there is a progressive shift towards an external perspective with the design and implementation of new management strategies, which are generally named with the term of supply chain management (SCM). Adopting a SCM strategy means to apply a business philosophy where more industrial nodes along a logistic network act together in a collaborative environment, pursuing common objectives, exchanging continuously information, but preserving at the same time the organisational autonomy of each single unit. However, in order to conduct this concept in practice, several hurdles are still to overcome (Samii 2004) (Hieber 2002), mainly due to:

- The conflicts resulting from local objectives versus network strategies, because supply chain is a multi decisional context, so companies must make decisions collectively.
- The difficulty in making decisions in a collaborative manner. It observed in several supply chain cases.
- The need for sharing sensitive information of participants in the SC. If the supply chain is composed by independent enterprises, sharing information becomes a critical obstacle, since each independent actor typically is not willing to share with the other nodes its own strategic data (as production capacity, internal lead times, production costs, sales forecasts, etc.).
- The need for sharing information technology tools.

In this paper, we focus on the modeling of agent-based distributed architecture of simulation in decision-making processes within the supply chain context. We propose mainly a set of negotiation protocols and model it within AUML modeling language.

The agent-based distributed architecture concerns two different kinds of simulation: classical simulation and proactive simulation. Classical simulation is a useful device to provide “what-if” analysis and to evaluate quantitatively benefits and issues deriving from operating in a cooperative and collaborative environment. It is a powerful technique to convince decision-makers to adopt a SCM process and to choose the most appropriate management strategies and practices for a given SC. Proactive simulation allows making decisions collectively in a short time (in particular, in case of disturbance). Also this distributed architecture allows simulating the future behavior of the supply chain participants.

This paper is structured as follows. In Section 2 we introduce the role of distributed simulation in the supply chain context. In Section 3 we show a generic model fixing and defining the group of participants in the supply chain. In Section 4 we present the class diagrams for agents. In Section 5 we propose a set of negotiation protocols and model it within AUML sequence diagrams. In Section 6 we present a case study which gives some tests of the negotiation protocols. Finally, we conclude the paper and give an overview of our future work.

THE ROLE OF DISTRIBUTED SIMULATION IN THE SUPPLY CHAIN CONTEXT

Many software vendors (e.g. IBM in (Bagchi et al. 1998)), universities and companies (Telle et al. 2003) (Banks et al. 2002) have traditionally used a local simulation approach in the supply chain context. Only in recent years, more and more companies in supply chain adopt distributed collaborative simulation (Brun et al. 2002), because it provides a connection between supply chain nodes that are geographically distributed throughout the globe, guaranteeing that each single simulation model is really linked to its respective industrial site. Moreover, companies in any supply chain must make decisions individually and collectively regarding their actions in *production, inventory, location, transportation and information* (Hugos 2003), then the distributed simulation can preserve at the same time the local autonomies and privacy of logistics data. In some cases, the execution of a distributed model allows to reduce the time spent for simulation, since separated models run faster than a single complex model (Fujimoto et al. 1999).

Despite the great use of simulation in SC and SCM, there are many additional opportunities for application of the methodology (Banks et al. 2002). However, many of these opportunities require that challenges be overcome (see Introduction). Aim of the agent-based distributed simulation presented in this paper is:

- To convince decision-makers to adopt a SCM process and to choose the most appropriate management strategies and practices for a given SC.
- To make decisions collectively in a short time. Mainly, in the case of operational planning (short term, for example: rush order) (Pinedo and Chao 1999) or in a situation where the supply chain partners negotiate a delivery dates modification due to a disturbance (for example: problem of production, problem of transport, etc.), because the decision system has to make its choice within a short time, and must be able to evaluate the consequences regarding various scenarios in distributed manner within a shorter time too.

In order to have a flexible and proactive model, we have chosen the Multi-Agent approach to develop our architecture. Clautier (Clautier et al. 2001), Maturana (Maturana et al. 1999) and Parunak (Parunak 1996) showed the main benefit to use this approach in the field of the supply chain. Thus, the complete architecture of simulation is made on a set of agents modeling the supply chain participants (figure 1). These collaborative agents communicate between them and negotiate using protocols. They seek the accurate and timely data that holds the promise of better coordination and better decision-making in the information systems of the supply chain participants (such as an ERP (Enterprise Resources Planning) system); this means every time the simulation starts, the model must be initialized with the currents states of the supply chain participants.

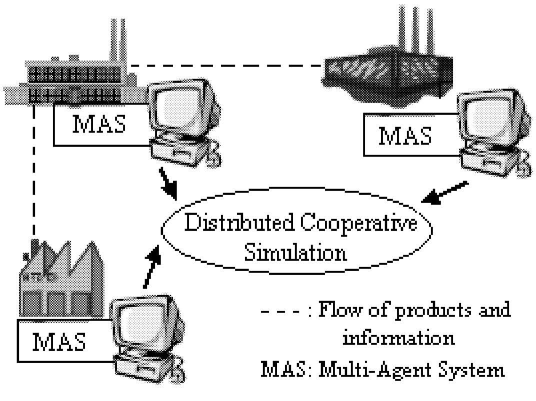

MAS
MAS
MAS
Distributed Cooperative Simulation
- - - : Flow of products and information
MAS: Multi-Agent System

Fig. 1 – Agent-Based Distributed Simulation for the SC

STRUCTURE AND PERIMETER OF SUPPLY CHAIN TO BE MODELLED

Two parameters are important in the process of the modeling of a supply chain, the perimeter and the structure. The first delimits the supply chain in a number of actors (companies), and the second defines the customer/supplier relationships. If the two aforementioned parameters are absent, it will be difficult to define the modeling boundaries (in term of levels). Indeed, a supply chain can use several tens, even hundreds of nodes geographically distributed throughout the globe. Then, is it really necessary to take into account all the actors? Moreover, a company can belong to several supply chains. Then, what are the levels of customers and suppliers that should be covered by the proposed model?

To answer these two questions, it is necessary to identify the product for which the supply chain is defined. Indeed, we define a supply chain for a product or a family of products. It is composed of all the companies involved in the design, production, and delivery of a product to market. Having a clear knowledge of the product, we better specify the central company of the supply chain, i.e. the one that assembles the finished product. Next, we propose to follow the steps below, they define a generic model for supply chain which delimits the boundaries of modeling and defines the customers/suppliers relationships:

Identifying the product:

1. Identify the finished product for which the supply chain is defined. This automatically defines the central company of the supply chain.
2. Identify the bill of materials of the finished product.
3. Exclude from this bill of materials the raw materials not requiring a partnership or collaboration.

Suppliers:

4. Afterwards, identify the remaining raw materials suppliers (distributors or factories having the activity of production). The same raw material can be bought from one or more different suppliers. In this last case, the percentage of the orders to place to each one of them should be determined.
5. For each supplier, the steps 2, 3 and 4 have to be renewed by considering, this time, the raw material as a finished product, and so on up to the upstream supplier.
6. Determine the type of the orders (stationary, random, etc.) of the customers (other than central company "EC") of the various suppliers identified in step 4.
7. Remake step 6 for the suppliers of the suppliers except for the last suppliers (upstream suppliers).

Customers:

8. Identify the list of customers of central company "EC".
9. In this list, identify the potential customers acting on the supply chain of the product (customers requiring collaboration or a partnership) and those which do not require the collaboration. Then, determine the type of the orders for the latter (random, stationary, etc.) as well as the percentage of the orders which each potential customer places to central company "EC".
10. For each potential customer of central company "EC", remake steps 8 and 9 except for the final customers (for example, consumers).

Agents-based Model:

11. Allot an Agents-based model (cf. Section below) to all the identified actors (central company, customers and suppliers).

AGENTS-BASED MODEL FOR SUPPLY CHAIN AND SUPPLY CHAIN MANAGEMENT

The shift from the proposed generic model to an agents-based model starts with the modeling of each actor (central company, customers and suppliers). To represent the three main functions of the company (*source*, *make* and *deliver*) and consider the control processes in the supply chain and its environment, each actor is modeled by seven agents, except the last suppliers and customers (retailers and consumers). These agents are: *AgentPRC* which plays the part of the processes related to the customer, *AgentDis* who manages the "stock of distribution", *AgentPro* which plays the role of the "make" process, *AgentApp* for the "source" process, *AgentAch* which plays the role of the "purchase" process, *AgentSCM* for the "management" of the SC and *AgentPer* which handles the "disturbances". This last agent makes it possible to the model to be open and extensible in order to consider a large variety of disturbances in order to cover various types of SC (world size, national size, branch of industry, etc.). In the case of an actor of the distributor type, the agents *AgentPro* and *AgentApp* do not exist.

In order to represent the relationships between agents and to define its elements (their attributes, operations, roles, protocols, etc.) we use the AUML class diagrams (Huget 2002). Figure 2 shows the conceptual level of the class diagram of an actor and figure 3 illustrates, as an example, the implementation level for the agent *AgentSCM.*

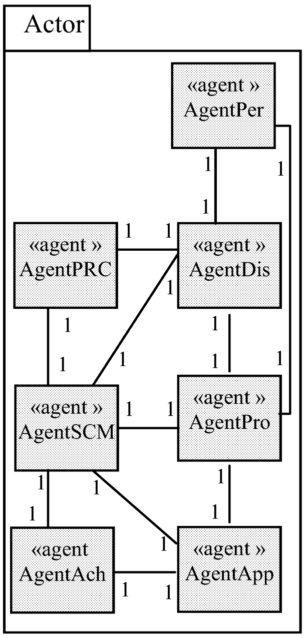


Fig. 2 - Class diagram: Conceptual level

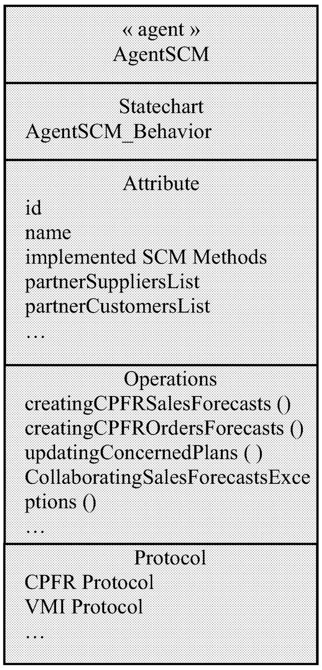


Fig. 3 – Implementation level for the "AgentSCM"

NEGOTIATION PROTOCOLS

The negotiation is the mechanism by which the agents can establish a common agreement. In the case of intelligent agents and of the MAS (Multi-Agent Systems), the negotiation is a basic component of the interaction because the agents are autonomous (Jenning et al. 2001); there is no solution imposed in advance and the agents must find solutions dynamically, while solving the problems. To model the negotiation between the agents composing our system, we consider the following aspects:

- *The negotiation object*: an abstract object which includes the attributes that the agents want to negotiate. In our architecture, several objects are prone to be negotiated according to the situation. We find among others, the Order and its attributes (quantity and delivery date), the Contract of Continuous Delivery and its attributes (quantities and plan of delivery), the Forecasts and their attributes (quantities, dates and exceptions) and the acceptable scenario in the case of dysfunctions.
- *The decision-making process*: this is the model that the agent uses to make the decisions during the negotiation. The most important part of making decisions is the negotiation strategy which allows the agent to choose the most appropriate communicative intention (also called "performative") at a certain time.The performative can be ACCEPT_PROPOSAL, REQUEST, INFORM, PROPOSE, etc.
- *The communication language*: the language used by the agents to exchange their knowledge and information during the negotiation. We use the FIPA-ACL language (FIPA 2002) in our application.
- *The negotiation protocol*: the set of elements that governs the negotiation such as the possible participants in the negotiation, the legal proposals that the participants can make, the states of the negotiation. And finally a rule to determine when the negotiation should be stopped in case of agreement (or when it is necessary to stop the negotiation process because no agreement could be reached).

In the SCM process, the agents are co-operative, having the same goal (aggregation of the local objectives). They share and solve problems together. For this reason, the agents must provide useful reactions to the proposals that they receive. These reactions can take the form of a counter-proposal (refused or modified proposal). A counter-proposal is an alternative proposal generated in response to a proposal. From such reactions, the agent must be able to generate a proposal which is probably ready to lead to an agreement. Consequently, the agents of our system must use protocols respecting the criteria which have been stated above and that mainly depend on three parameters:

- The branch of supply chain sector (textile and clothing sector, consuming goods sector, etc.);
- SCM strategies and practices used for the companies' co-operation and coordination;
- Objects to be negotiated: rush order, ordinary order, sales forecasts, orders forecasts; modification of delivery plans in case of trouble, etc.

We propose a set of negotiation protocols between agents peculiar to the supply chain management. They are:

- Protocols corresponding to the SCM strategies and practices (CPFR -Collaborative Planning, Forecasting and Replenishment-, Transshipment, etc.);
- Recursive heuristic negotiation protocol;
- Firm heuristic negotiation protocol.

We mainly focus in the following sections on the recursive and firm heuristic negotiation protocols, and the CPFR negotiation protocol used between two agents "AgentSCM" of a distributor and his supplier. We also model these negotiation protocols with AUML sequence diagrams (Huget and Odell 2004).

Heuristic Negotiation

The heuristic negotiation is shown in figure 4 (Florea 2002). In this protocol several proposals and counter-proposals can be exchanged in various steps. Agent "A", with proposal "pA", is the initiator of the negotiation, whereas the agent "B" (participant) can reply with the answers "p1B", "p2B" and "p3B" (to modify the request). The number of the counter-proposals is limited. Once this limit is reached, the agents arrive to a rejection. We propose to recapitulate the heuristic negotiation protocol using an AUML sequence diagram (figure 5).

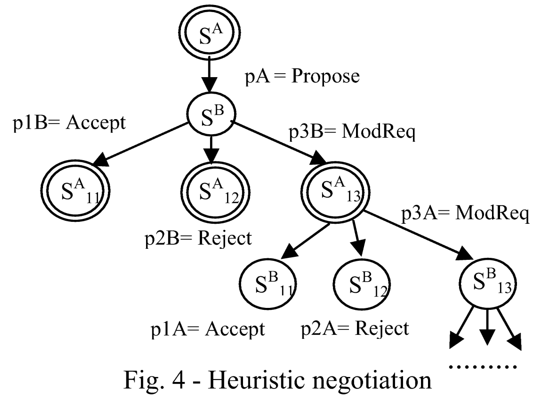


Fig. 4 - Heuristic negotiation

Proposal for a Firm Heuristic Negotiation

In some situations of negotiation, the cooperative agents must find an agreement; an example is the case of two agents "AgentSCM" that collaborate on the sales forecasts. For this reason, the heuristic negotiation (cf. figure 5) should include only ACCEPT-PROPOSAL or PROPOSE performatives (without the REFUSE performative). Thus, we propose the *firm* heuristic negotiation protocol which is a particular case of the heuristic negotiation. The word "firm" stands for this protocol since it always leads to an agreement. Figure 6 shows the sequence diagram that describes this protocol.

Proposal for a Recursive Heuristic Negotiation

The *recursive* negotiation protocol that we propose takes place at least between three agents, the initiator of the negotiation (sender), and the receiver who could become the initiator of a new heuristic negotiation with the third agent; hence the word "recursive" qualifying this heuristic protocol. Figure 7 shows the corresponding sequence diagram.

In our architecture of the simulation, the recursive heuristic negotiation either belongs to a protocol corresponding to a SCM practice and strategy or corresponds to the negotiation of a rush order or scenario to be adopted in the case of disturbance (production problems, disturbance of transport, etc.). In the general case, the negotiation takes place in the following way:

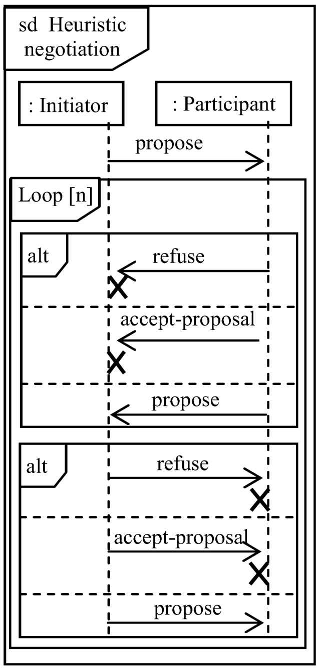


Fig. 5 - Heuristic negotiation

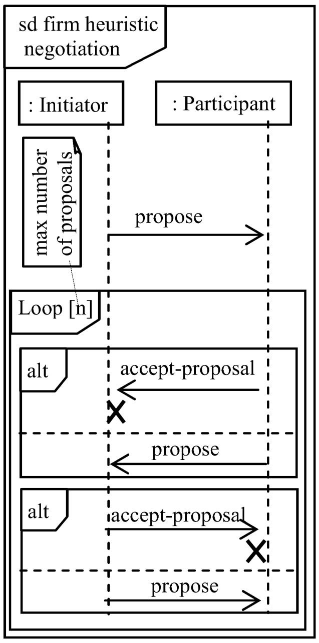


Fig. 6 – Firm Heuristic negotiation

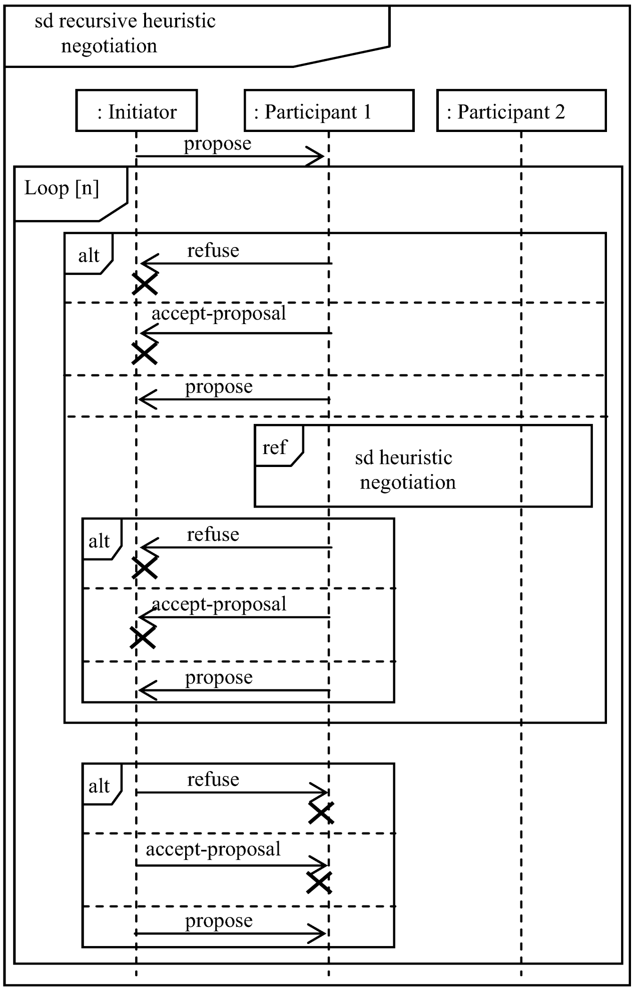


Fig. 7 - Recursive heuristic negotiation

- The initiator of the negotiation sends messages (not necessarily identical) of type PROPOSE to all the direct agents (upstream and/or downstream) whom he thinks could be candidates in a negotiation. So, the initiator launches several independent negotiations. It does not wait for all the answers to make a decision. Moreover, according to the situation and the time interval, it can come up with the best solution by creating new proposals deduced from the received answers;

- Since the agents of our architecture are co-operative, each one of them - receiver of a message - can start a negotiation if necessary between other agents in order to find the best solution.

CPFR Negotiation Protocol

The CPFR (Collaborative Planning, Forecasting and Replenishment) developed by the VICS Association (2006) (Voluntary Interindustry Commerce Standards) is a collaborative process that enhances VMI (Vendor Managed Inventory) and CRP (Continuous Replenishment Program) by incorporating joint forecasting. The CPFR negotiation protocol between a supplier and a distributor proceeds in the following way:

- The negotiation starts between the distributor agent “AgentSCM” and his supplier agent “AgentSCM” as soon as one of them creates sales forecasts and informs the other about them.
- The receiver agent consults and analyzes these forecasts, then sends a confirmation to the sender or initiates a *firm* heuristic negotiation with him if he does not agree in order to modify these forecasts. In all the cases, the sender and the receiver must agree on sales forecasts then share them.
- Each one of them can be the author of a heuristic negotiation which resolves an exception (changes/updates) concerning the created sales forecasts. This heuristic negotiation is prone to become *recursive* to involve other agents being able to contribute to the resolution. The negotiation finishes when all the exceptions are resolved or the dynamically fixed delay is expired.
- The agent “AgentSCM” of the distributor sends a message of type INFORM to the agent “AgentSCM” of its supplier containing information about the level of its stocks (orders quantities, inventory quantities…).
- The agent “AgentSCM” of the supplier also answers by a message of type INFORM indicating its capacity and history of production, its lead times…
- At this stage, the negotiation concerning the sales forecasts is finished. A new negotiation begins between the two agents (distributor agent “AgentSCM” and his supplier agent “AgentSCM”) as soon as one of them creates orders forecasts and informs the other about them.
- Each one of them can be the author of a heuristic negotiation which resolves an exception concerning the created orders forecasts. This heuristic negotiation is prone to become *recursive* to involve other agents being able to contribute to the resolution. The negotiation finishes when all the exceptions are resolved or the dynamically fixed delay is expired.
- Finally, according to the situation, one of them creates firm orders, and then immediately informs the other about them.

Figure 8 illustrates the corresponding sequence diagram when the supplier deals with the creation of the forecasts of sales and the firm orders.

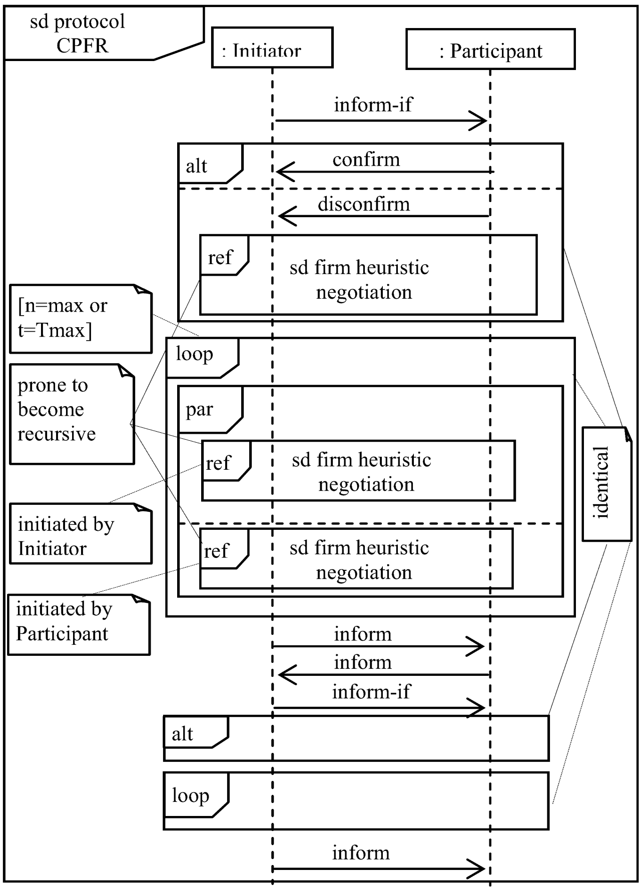


Fig. 8 – CPFR Protocol

CASE STUDY

To test the proposed protocols, we considered a particular case (similar to the industrial cases) of SC in the textile and clothing sector including a central company (head office) located in a country and of two subsidiaries F1 and F2 located in two different foreign countries. The functions of supplying raw materials, manufacturing and delivering to stocks abroad are carried out by the central company. The functions of sales and delivery to the wholesalers are done by the subsidiaries. We have supposed that F1 distributes the products to two competitor wholesalers, while F2 distributes the products to two non-competitor wholesalers located in two different geographical areas. Each of the two non-competitor wholesalers uses CPFR protocol to collaborate with its distributor (subsidiary company F2). However, the two competitor wholesalers use a simple collaboration without sharing their data with their same distributor (F1), because they fear that their strategic information would be diffused. The final customers (retailers and consumers) have been represented by an agent that places two types of orders to the wholesalers:

- Pseudo-cyclic orders corresponding to a P1 product.
- Uniform random orders related to the stage of new fashionable product marketing, P2.

During simulation, we have chosen the performance indicator “Number of Backlogged Delivery” for the two subsidiary companies. It represents an out-of-stock for the wholesalers. Table 1 shows the results obtained throughout the period of simulation.

Table 1: Performance Indicator

	Subsidiary F1		Subsidiary F2	
	P1	P2	P1	P2
Number of Backlogged Delivery	4	11	1	4

Results comment:

- As to the product P2, we notice that the subsidiary F2's backlogged delivery number is only 4 while F1's number reached 11. The difference between the two numbers is relatively important in this case. This shows that the F2 subsidiary and its wholesalers got profit from determined sales forecasts and orders forecasts since they implement the CPFR process -especially as the nature of the P2 product orders is random uniform-. This kind of orders is difficult to control, because the actors (Head office, Subsidiary companies and Wholesalers) do not have the entire ability to predict the overall amount of sales and orders within a given market. It is the case of several fashionable products in the textile and clothing sector.
- Concerning the product P1, we notice that the subsidiary F2's backlogged delivery number is only 1 while F1's number reached 4. The difference is not relatively large, because the P1 product orders are pseudo-cyclic. I.e. these variations' orders are well known. It is the case of several products in the textile and clothing sector whose orders vary according to seasons of the year (tee-shirt for summer and coat for winter).

We deduce that the CPFR process used by the F2 subsidiary and its wholesalers reduced considerably the number of backlogged delivery. In effect, the wholesalers' out-of-stock will be reduced. This will have a good impact on the quality of the offered service within the satisfaction of the final customer policy.

This case study, which aims only at testing the proposed protocols, shows that more the level of collaboration increases, the better the performance indicator is. The level of collaboration can increase by choosing supply chain management strategies and practices based on the sharing of data (like CPFR) and by integrating several actors in the decision-making.

CONCLUSION AND FUTURE WORK

In this paper, we presented the basis of a distributed architecture for proactive simulation to contribute to the collaborative decision-making in the supply chain. We proposed a generic model allowing a flexible modeling of the supply chain thanks to the multi-agent properties. The model, developed at this stage, makes it possible to test a set of negotiation protocols which we previously proposed and modeled with the AUML language. The given scenario shows that AUML diagrams offer effective solutions to specify and model real-world agent-based applications.

Considering the diversity of the platforms used by the industrial actors and the distributed nature of the SC, we have chosen the JADE development framework to develop the proposed agent-based distributed architecture.

The used tests are very promising and show that it is possible, by connecting the agents to the information systems of supply chain actors (APS or ERP, etc.), to bring an important help to the collaborative decision-making.

In the next stage of this work we will validate and enhance the components of the proposed architecture on some industrial cases from the textile and clothing sector.

REFERENCES

Bagchi, S. Buckley, S. Ettl, M. and Lin, G. 1998. "Experience using the IBM supply chain simulator". *In Proceedings of the 1998 Winter Simulation Conference*, Washington, D.C, USA, pp: 1387-1394.

Banks, J. Jain, S. Buckley, S. Lendermann, P. and Manivannan, M. 2002. "Panel Session: Opportunities for Simulation in Supply Chain Management**".** *In Proceedings of the 2002 Winter Simulation Conference*, San Diego, California, pp: 1652-1658.

Brun, A. Cavalieri, S. Macchi, M. Portioli-Staudacher, A. and Terzi, S. 2002. "Distributed simulation for supply chain co-ordination". *In Proceedings of the 12th International Working Seminar on Production Economics, Igls,* Austria.

Cloutier, L. Frayret, J-M. D'Amours, S. Espinasse, B. and Montreuil, B. 2001. "A commitment-oriented framework for networked manufacturing coordination" *International journal of computer integrated manufacturing*, 14(6): 522-534.

FIPA, 2002. "FIPA ACL Message Structure Specification". Foundation for Intelligent Physical Agents. (on http://www.fipa.org/spcs/fipa00061/SC00061G.pdf).

Florea, A. 2002. "Using Utility Values in Argument-based Negotiation." *In Proceedings of IC-AI'02, the 2002 International Conference on Artificial Intelligence*, Las Vegas, Nevada, USA, pp: 1021-1026.

Fujimoto, R. 1999. "Parallel and distributed simulation". *In Proceedings of the 1999 Winter Simulation Conference*, IEEE, Piscataway, NJ, pp. 122–131.

Hieber, R. 2002. "Supply chain management: a collaborative performance measurement approach" VDF (eds.).

Huget, M-P. 2002. "Agent UML Class Diagrams Revisited". *In Proceedings of Agent Technology and Software Engineering (AgeS)*. Bernhard Bauer, Klaus Fischer, Joerg Mueller and Bernhard Rumpe (eds), Erfurt, Germany.

Huget, M.-P. and Odell, J. 2004. "Representing Agent Interaction Protocols with Agent UML". *In Proceedings of the Fifth International Workshop on Agent-Oriented Software Engineering* (AOSE 2004), Paolo Giorgini, Joerg Mueller and James Odell (eds.), New York.

Hugos, M. 2003. "Essentials of Supply Chain Management". *John Wiley & Sons (eds.),* pp: 5-6.

Jennings, N. R, e.a. 2001. "Automated negotiation: Prospects, methods and challenges." *Group Decision and Negotiation Journal,* 10(2), pp: 199-215.

Maturana, F. Shen, W. and Norrie, D. 1999. "Metamorph: an adaptive agent-based architecture for intelligent manufacturing." *International Journal of Production research,* 37(10), pp: 2159-2173.

Parunak, H. V. D. 1996. "Applications of distributed artificial intelligence in industry." *In O'Hare, G. M. P. And Jennings, N. R., editors, Foundations of Distributed Artificial Intelligence, John Wiley et Sons (eds.),* pp: 71-76.

Pinedo, M. and Chao, X. 1999. "Operations Scheduling with applications in manufacturing and services." *Mc. Graw-Hill (eds.).*

Samii, A-K. 2004. "Stratégie Logistique, Supply Chain Management" *Dunod (eds.),* 3rd edition, pp:14-15.

Telle, O. Thierry, Caroline. And Bel, G. 2003. "Simulation d'une Relation Client/Fournisseur au sein d'une Chaine Logistique Intégrée: Mise en Oeuvre Industrielle". In *Proceedings of MOSIM03 (Conférence Francophone de MOdélisation et Simulation),* Toulouse, France.

VICS Association, 2006. *Web site* (last access: May 2006).

AGENT BASED VS NESTED SIMULATION FOR SUPPORTING ON-LINE TELLER SCHEDULING IN GROCERIES SUPERMARKET DISTRIBUTION: A CASE STUDY

Roberto Revetria
CIELI, Centro Italiano di Eccellenza sulla Logistica Integrata
Via Bensa, 1
16126 Genova GE
roberto.revetria@unige.it

Cinzia Forgia, Alessandro Catania
IQR Consulting, Mindrevolver.com SaS Group
Via delle Medaglie d'Oro, 96/2
17031 Albenga SV, Italy
{cinzia.forgia, alessandro.catania}@mindrevolver.com

KEYWORDS: Discrete Simulation, Retail, SIMULA, Nested Simulation

ABSTRACT

Modeling and simulation has been widely used for supporting the teller planning in supermarket chains where it plays a crucial role in predicting customer waiting times and queues length according to the number of tellers opened. This approach requires the accurate modeling of the customer behavior at the single entity level. Such approach is based, however, on some assumptions made on the behavior of the customers during purchasing activity that is often very hard to be obtained, requiring long data collection campaign that prevent the methodology to be used for hour by hour scheduling. The proposed approach demonstrate as on-line simulation could be effectively used to support next period queues length and customers' waiting time by implementing a SIMULA nested simulator able to reproduce both the "Real World" and the "Virtual World". Proposed approach can be applied also to other real life application where there is a lack of knowledge in the behavior of the single entities such as highway toll plaza, shops and fast foods chains. The paper present the methodology, a real life application and results discussion.

INTRODUCTION

There is a wide class of phenomena involving intelligent agents (i.e. customer behavior in a shop, drivers at a toll plaza) that can be accurately modeled only by having an exact knowledge of the internal agent's behavior. Such knowledge can be obtained at the price of long data collection campaign, multiple agent tracing, social behavior modeling and many other time consuming techniques.

In the supermarket distribution industry there is a growing interest in the possibility of accurate modeling the behavior the customer in order to optimize the number of tellers to be opened in each period or the best way to use workers for shelves replenishment.

In a supermarket, in fact, a stock out event is generally associated with a potential lost sale whose criticality is proportional to the duration of the stock out and the number of the items involved. Another important issues is related to the average time spent by a single customer into the shops, is not a secret that long waiting time in the teller queue seriously affect the profitability of the shop. This point is becoming more important in some countries, like Italy, where a significant portion of the fresh product are sold by operators inside special area of the supermarket.

In Italy, in fact, customer perception of the quality of the same raw ham pre-packed in self-service or sliced-at-the-moment could be dramatically different. Resulting for a only self service supermarket a consistent sale loss at the short term and a consequent customer loss in the medium term.

In the recent years several techniques have been used to trace customer behavior into shops ranging from RFID tag on the trolley up to dedicated scanning camera placed is selected points of the shops. The results where generally poor: since the natural variability of the process drives the observation very quickly out of control.

Another important issues is related to external driven variability: in bad weather days the number of customer spending their off-duty afternoon in supermarket or department store increase dramatically, again there is a grate interest in supermarket industry in forecasting queues length and customers' waiting times according to the weather change.

Since many agent based simulation project failed in the past for the lack of credible data in the customer behavior modeling a different approach should be used.

In the proposed approach we abdicated the agent based behavior-modeling chimera in favor of a quasi-black box approach where some parameters can be actively measured and used for a input/output model.

In real life application a supermarket can be summarized in a black box where customers enter at the entrance gate, spend some time in purchasing - according on their internal needs – and present themselves to the tellers for paying. While is quite simple to measure the input rate, is very hard to investigate on the single entity choice. The other important measures that can be conducted on the supermarket are: the post teller output rate and the check composition. While the first is only a mediated measure of the teller efficiency the second could be used for obtaining an impressive list of information such as estimation of the average time spent in purchasing and/or in internal queues.

At this point a quasi-black box simulator can be used during regular operating hours to investigate the present system for possible evolution and to choose the best performing tactics. Since the continuous updating of the simulator an evolving scenario can be easily accommodate.

NESTED SIMULATION

The use of on line simulation for supporting complex decision making in evolving scenario has been widely documented in several industrial application, however the great part of those was mainly related to manufacturing industry where the extensive use of automation transform the data collection into a matter of software integration. In supermarket industry is simply impossible to track customer behavior without afford high cost and without violate the privacy protection act.

It is not possible to study purchasing behavior of the customer at entity level so is very difficult to implement an agent based simulation in order to support the teller opening schedule.

On the other side the use of the quasi-black box simulation methodology requires a credible validation in order to be fine tuned and adopted into a real life application. The needs of a test bed for the quasi-black box methodology requires the adoption of a model able to reproduce an internal simulation and its decision making process while simulating normal operation procedure and scenario evolution.

The author addressed this issue to the use of nested simulation where the outer model will be used to simulate the real world and the nested simulation will be used to act as the on-line simulator and its consequent Decision Making Process. The complete schema is presented in figure 1.

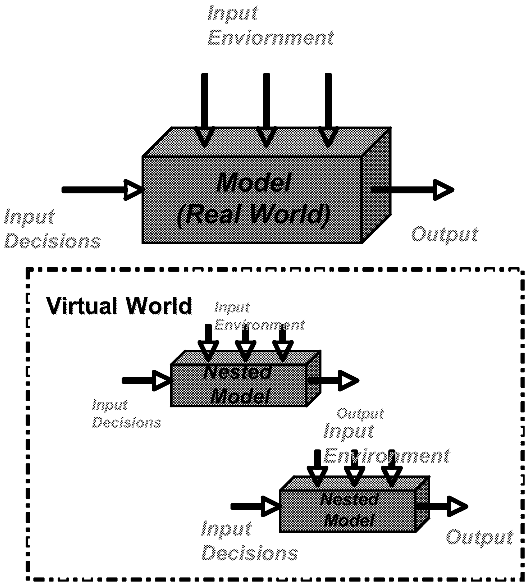


Figure 1: The Use of Nested Simulation in the Quasi-Black Box Approach

In the proposed application a simple supermarket model is modeled in the outer simulator, the customer arriving process is modeled by using a uniform distribution of the mean time between arrivals. A multiple period simulation exercise is the used to simulate up to 9 different customer arriving rate during a single day. In this way a possible perturbation path can be actively modeled by continuously adjusting the arriving rate.

In the same simulator a nested one is implemented using the same logic but obtaining its distribution data from statistics of the outer simulator. Practically the inner simulator can obtain the customer arriving ratio and the customer mean processing time from sample made in the “Real World”. Like in reality the simulator can only investigate the possible evolution of the “Real World” from the incomplete samples “visible” as stated in figure 2.

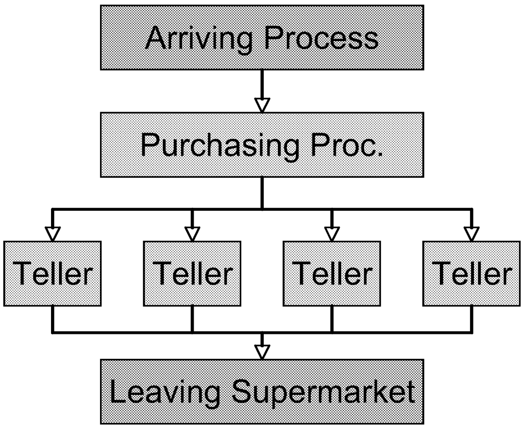


Figure 2: Supermarket Observable Process

In other world is visible only the Arriving Process and the Leaving Supermarket process and all the other information can only be guessed.

In this way the inner simulation has the visibility on the real world similar to the one that it will have in the real application.

During the simulation of the “Real World” the user will be prompted for a choice made on the possible scenario evolution computed from the present point and according to a predefined teller opening schema. This fact leads to multiple time axes (see fig. 3) departing from the “Real” to the various “Virtual” worlds. Since internal statistics are continuously updated the inner simulation can be used to investigate the possible reaction to an evolving scenario.

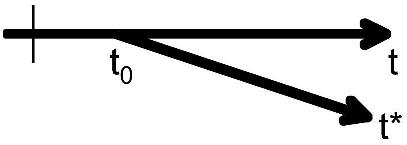


Figure 3: Virtual World Time Axes (t) Emerging from Real World (t) at Time t_0*

The nested simulation model can now be applied to the quasi-black box methodology to test the effectiveness of the approach.

SIMULA IMPLEMENTATION

Among the various tools and language suitable for simulation application the authors decides to implement the technological demonstrator using SIMULA a general-purpose language with a specific capability for nested simulation. This choice was driven following the literature that indicate in SIMULA and in Java (Kindler et al. 1997) the two languages able for supporting complex exercises in nested simulation.

SIMULA is today available and supported in several commercial implementation, however the authors decide to use GNU Cim version 3.36.

GNU Cim is a compiler for the programming language SIMULA (except unspecified parameters to formal or virtual procedures (It offers a class concept, separate compilation with full type checking, interface to external C routines, an application package for process simulation and a coroutine concept.

GNU Cim is a SIMULA compiler whose portability is based on the C programming language. The compiler and the run-time system is written in C, and the compiler produces C code, that is passed to a C compiler for further processing towards machine code.

GNU Cim is copyrighted by Sverre Hvammen Johansen, Stein Krogdahl, and Terje Mjs, Department of Informatics, University of Oslo.

The implemented model followed a two step approach: on the first step a simple purchasing-teller model was created and tested for nested simulation capability proving the potential of the methodology. The customer enter into the system with a random uniform distribution, he now spend another time in purchasing and present himself to the first available teller which queue is the shortest among the available. The number of active tellers are defined at the beginning of the 9 simulation periods, customers eventually left in a closing teller are served before closing. The outer simulator collect statistics on the arriving customers (minimum and maximum inter arrival time) and use it for the internal simulation.

The implementation was tested on a Mac OS X 10.4.7 1 GHz PowerPC G4 with 1.25 GB DDR SDRAM.

The simulation tested the various scenario keeping open 1 to 6 teller. At each simulated step the user was prompted for the opening teller choice, with the results of the simulation made in the "Virtual World" and immediately informed about the result of the simulation in the "Real World". As is possible to see in figure 4, the nested simulation model was able to reproduce the path of the real workload in its internal simulation allowing the user to make the right choice.

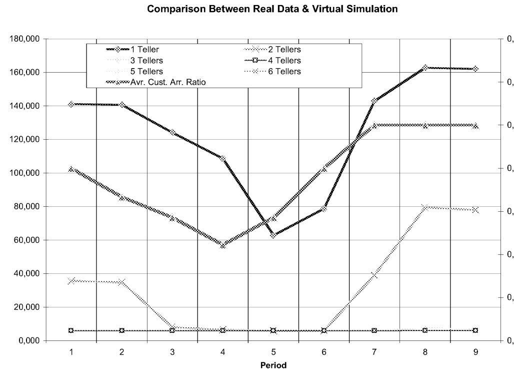


Figure 4: Comparison among Real World and Virtual World Workload

The use of SIMULA pose some difficulties in Verification & Validation, such language, in fact, lack of the modern tool features for rapid GUI development, this is becoming very critical when compared to Arena™, Simul8™, ProModel™ tools that have nice tool for supporting simulation visualization.

In order to avoid this problem, authors used Wolverine Proof Animation™ that support very hi fidelity visualization for general purpose simulation language. Proof Animation™ provides both off-line and on-line animation on Windows platforms and can be used by a wide variety of programs.

Second implemented model introduced a more complex behavior of the modeled customer that now has the ability of moving to an adjacent teller queue if perceived enough shorter than the one is currently placed. This can lead to an interesting behavior especially when a new opening teller position became available in peak period. The model was compared to an existing Simul8™ Supermarket model previously validated on a real life application. In figure 5 the Mean Squared Pure Error of the two models are compared showing an equal path in predicting the normalized Customer Served per minute.

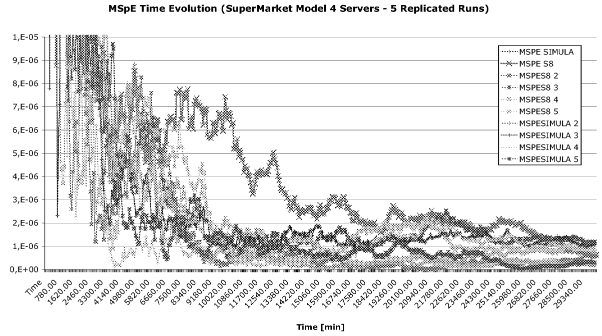


Figure 5: MSpE Comparison between SIMULA Model and Simul8 Previously Validated Model

The effect of the queue shifting algorithm implementation is visible in figure 6, the SIMULA model implement it while the Simul8™ does not. In the MSpE time evolution is possible to notice that unbalanced teller queue lead to a higher Mean Squared Pure Error since the newly available queue cannot affect the workload of the previously working leaving the potential of the system into a smaller number of resources. Such event is then disappearing when the new opened tellers reach their regimen.

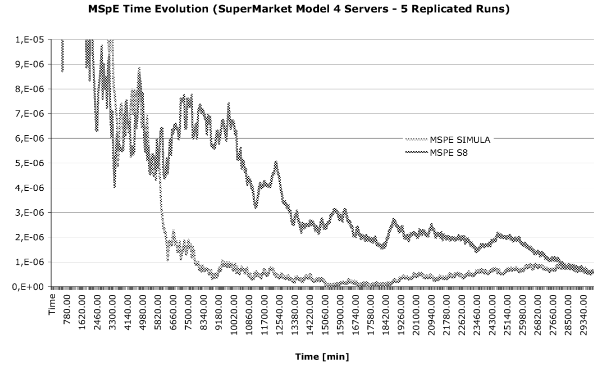


Figure 6: MSpE Effect of the Queue Balancing Algorithm

Final implemented model introduced an internal Fresh Product shop inside of the supermarket area, now customers have the chance to buy some fresh product directly into the supermaket and are served by a two server position that generate an internal queue system.

Again the internal simulation, representing the "Virtual World", is driven by statistics generated by the outer simulation model, representing the "Real World". Control point of this new model are the customer arrival, obtained at the customer's entrance metering system, and the total percentage of customers spending their time into the Fresh Product Store obtained with the customer check analysis.

EXPERIMENTAL CAMPAIGN

Proposed application was tested with a complete implementation including Fresh Product internal store, 15 tellers row and a queue-balancing algorithm.

The simulator provide a 12 hour period covering the typical opening hours of an Italian Supermarket (9AM-9PM), the evolution of the interarrival customers' time the following model was implemented.

$$\frac{1}{\Lambda_M}(t)=\frac{1}{\lambda_{Max}}+\left(\frac{1}{\lambda_{Max}}-\frac{1}{\lambda_{\min}}\right)\cdot\sin\left(\frac{t}{T}\cdot\pi\right)$$
$$\frac{1}{\Lambda_m}(t)=2.0\cdot\frac{1}{\Lambda_M} \qquad (1)$$

where:

- $\lambda_{\min}$:is the minimum interarrival rate (off hours) and is measured in customer per minute;
- λ_{Max}:is the maximum interarrival rate (peak hours) and is measured in customer per minute;
- Λ_M: is the maximum generated interarrival time (peak hours);
- Λ_m: is the minimum generated interarrival time (off hours);

Within this model the peak hours are from 11AM to 4 PM that is the usual busiest period in a Italian supermarket.

Implemented model requires only a data collection for the teller process and the internal Fresh Product store, both internal (nested) and external (outer) simulation models works on the same statistics implementing a Standard Random Generator for the teller process and a Uniform Random Generator for the internal Fresh Product Store.

The internal simulation serves all the possible configuration from 1 to 15 opened tellers driving the possibility to investigate how many tellers keep opened in each of the 12 hours of the day shift.

Since there is no knowledge about the number of customer in queue at the end of each "Real World" hour, the internal simulator may start with a warm up period of 15 minutes. Investigations made with a complete vision of the queue from the internal simulation demonstrate that the effectiveness of this choice was widely proven.

The simulation results are presented in the following table demonstrating the effectiveness of the proposed methodology, in figure 7 the comparison among Virtual Time in System and Real Time in System is Presented.

Period	Real World TIS	Real World NCS	Tellers	Virtual World TIS
1	30,9963	69	15	30,9963
2	42,1135	166	12	36,2361
3	53,8603	242	9	31,5624
4	53,0963	278	4	30,7059
5	51,6303	297	3	29,0502
6	50,4451	312	3	28,8962
7	49,4897	326	3	29,3479
8	48,7225	339	3	28,54170
9	47,9073	355	3	29,0987
10	47,1439	372	4	29,5933
11	46,7225	394	5	29,4638
12	45,0198	425	6	30,4355

Table 1: Comparison among Virtual and Real Simulation Result, Raw Data

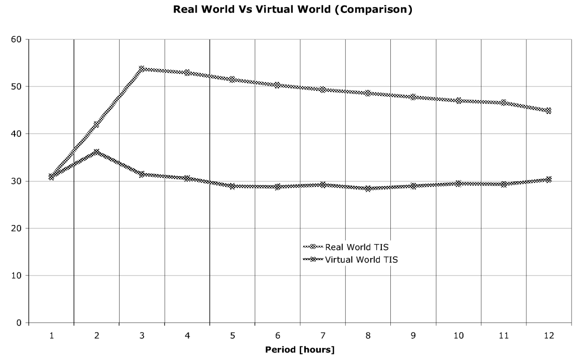


Figure 7: Comparison among Virtual and Real Simulation Results

CONCLUSIONS

The paper presented an application of the nested simulation to support supermaket operation and teller schedule optimization applied to a real case.

Proposed methodology can be adapted to several other cases (i.e Highway toll plaza) where an on-line simulation could help managers to take better decision. The use of the quasi-black box method can reduce the data collection time and costs resulting in a wider applicability of the nested simulation principle.

The author are now applying such methodology in combination with the Wearable PC technology into a project proposal for supporting the logistics operations into a real Supermarket Chain.

REFERENCES

AA.VV. Sociologia della Comunicazione Anno XIII n° 25 ed. Franco Angeli

Bruzzone A., Mosca R., Orsoni A., Revetria R. (2001) "Forecasts Modelling in Industrial Applications Based on AI Techniques", International Journal of Computing Anticipatory Systems (extended from Proeedings of CASYS2001, Liege Belgium

August 13-18), Vol 11, pp. 245-258, ISSN1373-5411

Bruzzone A.G., Revetria R. (1999) "Artificial Neural Networks as Support for Logistics in Super-Market Chains", Proceedings of HMS99, Genoa, September 16-18

Bruzzone A.G., Revetria R., Brandolini M., Massei M., Simeoni S. (2004) "Models for the Introduction of Mobile Technologies in External Logistics", Proceedings of ASTC2004, Arlington VA, April;

Bruzzone Agostino, Revetria R., Genovese M., Rombi L. (2001) "Inventory Management by Integrating ERP Systems and Simulation Tools", Proceedings of SCI2001, Orlando, July 22-25

Eugene Kindler: (2004) SIMULA and Super-Object-Oriented Programming. Essays in Memory of Ole-Johan Dahl 2004: 165-182

H.C. Weizmann J.K Gestione delle Risorse Umane e Valore dell'Impresa. Weizmann ed. Franco Angeli

L. Gallino L'impresa irresponsabile Gli struzzi ed. Einaudi

Peter Blümel, Eugene Kindler: (1997) Simulation of Antagonist Mutually Simulating Systens - First Experiences. SimVis 1997: 56-65

Revetria R. (2001) "Replenishment Policy Optimization Using the Simulation and the Fuzzy Logic Approach" Proceedings of The SCI2001 Conference, Orlando FL, July 22-25

Excerpt 1: NS2 Model User Interaction

```
MacRevetriaR:~/documents/downloads/simula/simprogs
robertorevetria$ ./NS2
  *** Results of internal simulation *** Period 1
      Tellers     Average time spent
            1                141.242
            2                 35.632
            3                  6.064
            4                  5.939
            5                  5.939
            6                  5.939
Enter the number of tellers :
4
  *** Report on external simulation ***
    47 customers ready at time     120.00
Average time in system:       5.92
Maximum time in system:       8.26
  4 total Operators Used
Press Enter to Continue.

  *** Results of internal simulation *** Period 2
      Tellers     Average time spent
            1                140.749
            2                 34.649
            3                  6.053
            4                  5.939
            5                  5.939
            6                  5.939
Enter the number of tellers :
4
  *** Report on external simulation ***
    88 customers ready at time     240.00
Average time in system:       5.95
Maximum time in system:       9.44
  8 total Operators Used
Press Enter to Continue.

  *** Results of internal simulation *** Period 3
      Tellers     Average time spent
            1                124.150
            2                  8.148
            3                  5.927
            4                  5.924
            5                  5.924
            6                  5.924
Enter the number of tellers :
3
  *** Report on external simulation ***
   124 customers ready at time     360.00
Average time in system:       5.98
Maximum time in system:       9.44
 11 total Operators Used
Press Enter to Continue.

  *** Results of internal simulation *** Period 4
      Tellers     Average time spent
            1                108.643
            2                  6.360
            3                  5.955
            4                  5.954
            5                  5.954
            6                  5.954
Enter the number of tellers :
3
  *** Report on external simulation ***
   152 customers ready at time     480.00
Average time in system:       5.99
Maximum time in system:       9.44
 14 total Operators Used
Press Enter to Continue.

  *** Results of internal simulation *** Period 5
      Tellers     Average time spent
            1                 62.802
            2                  5.954
            3                  5.954
            4                  5.954
            5                  5.954
            6                  5.954
Enter the number of tellers :
2
  *** Report on external simulation ***
   186 customers ready at time     600.00
Average time in system:       6.16
Maximum time in system:       9.76
 16 total Operators Used
Press Enter to Continue.

  *** Results of internal simulation *** Period 6
      Tellers     Average time spent
            1                 78.727
            2                  5.982
            3                  5.919
            4                  5.919
            5                  5.919
            6                  5.919
Enter the number of tellers :
3
  *** Report on external simulation ***
   231 customers ready at time     720.00
Average time in system:       6.08
Maximum time in system:       9.76
 19 total Operators Used
Press Enter to Continue.

  *** Results of internal simulation *** Period 7
      Tellers     Average time spent
            1                142.869
            2                 38.989
            3                  6.065
            4                  5.944
            5                  5.944
            6                  5.944
Enter the number of tellers :
4
  *** Report on external simulation ***
   290 customers ready at time     840.00
Average time in system:       6.05
Maximum time in system:       9.76
 23 total Operators Used
Press Enter to Continue.

  *** Results of internal simulation *** Period 8
      Tellers     Average time spent
            1                162.872
            2                 79.565
            3                  7.444
            4                  6.015
            5                  5.986
            6                  5.986
Enter the number of tellers :
5
  *** Report on external simulation ***
   349 customers ready at time     960.00
```

```
Average time in system:       6.05
Maximum time in system:       9.76
 28 total Operators Used
Press Enter to Continue.

  *** Results of internal simulation *** Period 9
      Tellers     Average time spent
            1               162.077
            2                78.005
            3                 7.132
            4                 5.999
            5                 5.981
            6                 5.981
Enter the number of tellers :
5
  *** Report on external simulation ***
   405 customers ready at time    1080.00
Average time in system:       6.04
Maximum time in system:       9.76
 33 total Operators Used
Press Enter to Continue.
```

CROWD AND GROUP SIMULATIONS

AGENT BASED SIMULATION ARCHITECTURE AUGMENTED BY ACTORS

Norbert Adamko
Valent Klima
Faculty of Management Science and Informatics
University of Zilina
Slovak Republic
E-mails: Norbert.Adamko@fri.utc.sk, Valent.Klima@fri.utc.sk

KEYWORDS

Agent based simulation, actors, simulation architectures, service systems modelling.

ABSTRACT

The paper deals with the agent based architecture ABAsim, which provides a solid base for creation of flexible, open and maintainable simulation models of complex service systems. In original ABAsim architecture, simulation models are composed of cooperating agents, which are organised in a hierarchical structure. To guarantee flexible modelling of service systems with intelligent and autonomous entities, ABAsim architecture has to be augmented with elements, based on the paradigm of actors. In the proposed architecture, actors are considered to be agents of lower order and are controlled by agents. Behaviour of actors is goal driven; they can communicate with agents via messages and are able to interact (not using messages) with other actors and entities. Changes required to integrate actors into the existing architecture design are also discussed.

INTRODUCTION

Architecture *ABAsim* was developed mainly for simulations of large service systems. A *service system* is understood as a system focused on elaboration of *orders* (attendance of *customers*) and execution of *services* related to them. The mentioned services can further initiate another set of orders. Service systems include a wide class of various systems e.g. factories, any transportation and logistic nodes/junctions, hospitals, repair shops etc. Natural and technical systems do not belong to that domain.

Design of the ABAsim architecture was during its development influenced mainly by the need for creation of complex models of transportation systems (e.g. marshalling yards or factory sidings), which to some extent also pre-determined some of the architecture properties. Let us mention the most important features of transportation service systems, which essentially influenced the original properties of the architecture:

- The service system structure can be considered (from the viewpoint of order elaboration) as strictly *hierarchical*. The order (the customer) entering the system (e.g. sort a train on the hump) calls recursive sequence of suborders according to the rules of competence redistribution.
- All system elements (subsystems) work synergically, unlike the majority of natural systems, with the common goal to elaborate the order. Thus, the architecture was not determined to work with the processes of the following kinds: evolution, competition and parasitism.
- The entities of a service system (orders/customers and resources) can be divided into specialized classes with the same behavioural rules for included entities (e.g. shunting locomotives in a marshalling yard). It means that the responsibility for the behaviour of entities is taken by their superior subjects (agents) and hence there is no reason to consider entities as agents.

ABASIM PROPERTIES

The ABAsim (Agent Based Architecture of simulation models) provides means for design of flexible simulation models of complex service systems. In the following text we will try to explain only the basic properties of the ABAsim architecture, more in-depth explanation can be found in Adamko et al. 2004.

In the ABAsim architecture, simulation models are composed of cooperating agents, which are organised in a hierarchical structure. As already mentioned, hierarchical structure is a typical property of transportation systems and many other service systems.

The simulation models of simple systems could be composed of only one agent; however the simulations of complex service systems are obviously connected with multi-agent approach using the agents within some organizational structures. One of the goals of the architecture is to try to model not only the behaviour of the modelled system but also its internal structure as close as possible.

Each agent is composed of internal components responsible for sensorial, executive and communication activities of agent. Each agent contains one *manager* (responsible for control, decision making and communication tasks) and a set of *assistants*. Agent's assistants could be divided to *instant assistants* (to this group belong *action*, *query* and *advisor*) and *continual assistants* (*process*, *monitor* and *scheduler* are part of this group). Based on the roles of assistants, we can divide the assistants into three groups, each group containing one instant and one continual assistant: *solvers* (advisor, scheduler), *effectors/actuators* (action, process) and *sensors* (query, monitor). Continual assistant execution is invoked strictly only by agent's manager.

All communication, internal inside-agent (between manager and other components) and also external inter-agent, is realised exclusively by messages. Following list contains all message types used in ABAsim architecture:

a) Inter-agent communication:
   - *Notice* – contains some information for the addressee without expecting any answer.
   - *Request* – carries specific demands which are expected to be satisfied or supplied.
   - *Response* – represents a reply to a Request-message (sent exclusively to the initial Request-sender).

b) Messages sent by managers to assistants:
   - *Start* – initiates an autonomous operation by a continual assistant, which can communicate only with the superior manager.
   - *Break* – represents the only way the manager can influence the autonomous run of a continual assistant, i.e. incontrovertibly terminate its operation. That kind of message is utilized only in an exceptional case under special conditions where there is no sense in continuing the operation of a continual assistant.
   - *Execute* – is a specific message through which the manager promptly mines required results from instant assistants.

c) Assistant messages:
   - *Finish* – represents compulsory notification, which has to be sent to the superior manager, about the fact that a continual assistant has just finished its operation.
   - *Notice* – sent to the superior manager any time during an assistant‘s operation, this provides information about some important facts or situations which have occurred.
   - *Execute* – can be utilized for the same purpose as those mentioned above, i.e. for mining results from instant assistants.
   - *Hold* – represents the only way to augment simulation time. The *time stamp* of the message defines the duration of its delivery, which can be equal or greater than the current simulation time. The continual assistant sends this message to itself with some time delay.

Modelled system is in ABAsim architecture represented by permanent hierarchical structure of agents, which are responsible for their respective tasks. The agent structure does not change during execution of the model, no new agents are created during execution of the model nor are any agents released from the structure. As already mentioned, single entities are not represented by agents (as many other architectures do), but agents manage many entities, which share the same behaviour. Model entities (customers, resources) are considered to be unintelligent, without any initiative and are only manipulated by agents' effectors.

ABASIM MODEL STRUCTURE

To demonstrate a typical structure of simulation model designed under ABAsim architecture we will utilise an example model of simple service system, in which customers enter a service hall, they move inside the hall and are served by two different kinds of resources, after they have been served they leave the hall. In this example model we are not interested in detailed modelling of customers' movement, let's say they move on straight lines from one place to another by specified walk speed.

On the top of the hierarchy, there is always single agent responsible for the whole model – *Model Agent*, called also *boss* of the model (Fig. 1). The fact that we can distinguish between surroundings of the hall and the hall itself is reflected also in the hierarchy of agents, therefore *Model Agent* has two subordinated agents responsible for respective parts of the system. *Surroundings Agent* is responsible for arrival of customers from surroundings (customer arrival times can be generated using random number generator) and departure of customers from the system (served customers leave the hall and are released).

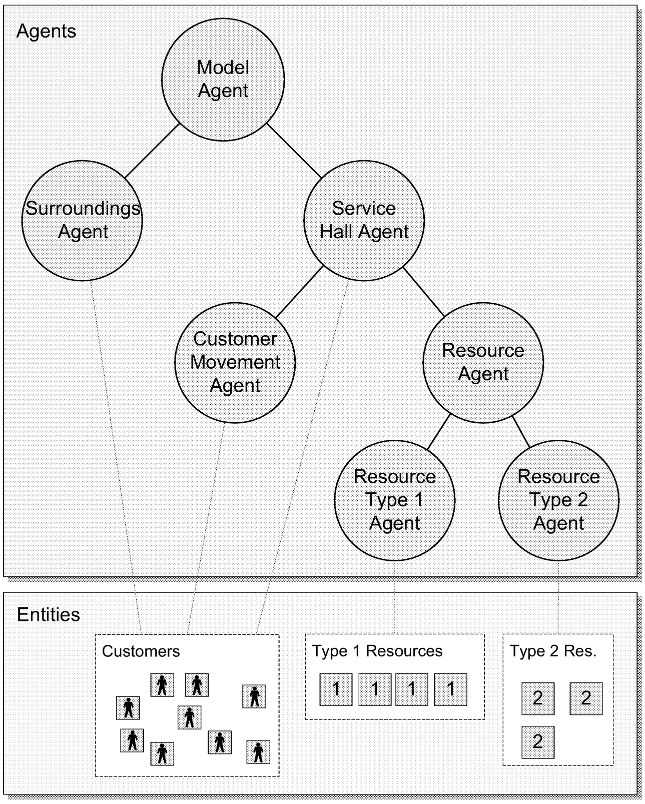


Figure 1: Structure of service system simulation model (original architecture)

Service Hall Agent manages the service of the customers. Each customer after entering the hall wants to be served by two different resources, first by resource of Type 1 and then by resource of Type 2. After the customer has been served by both types of resources, it leaves the hall. *Service Hall Agent* does not execute all service activities by itself; it delegates its duties and responsibilities to subordinated agents, which are responsible for resources (*Resource Agent*) and movement of customers (*Customer Movement Agent*). Anytime a customer requires a resource *Service Hall Agent* asks *Resource Agent* (by sending a *Request* message) to provide specified resource type. *Resource Agents* delegates the management and control of resources of different types to specialized agents (after receiving a request, *Resource Agent* determines, which of its subordinated agent is responsible and then passes the requests to it). Each of these resource agents controls all resource entities of the specified type (resources of the same type share the same, well defined, behavioural rules). Movement of customers between various places inside a hall is controlled by *Customer Movement Agent*. Since we require only simple modelling of customers' movement, without any interactions between them, we can very conveniently control a whole group of customer entities by single agent.

The structuring of a simulation model as well as of individual agents represents such features, which highly support the flexibility of simulation model within ABAsim architecture, among other features it allows changing a sub-model structure without the need of additional changes within other model parts. The reasons, why to change a sub-model structure, can be motivated e.g. by an additional decision to implement that sub-model on the more detailed level. For example, in the mentioned service system model example, we can decide to precisely model movement and interactions between customers (e.g. we could be also interested in examination of hall emergency exit design in regard to crisis situations). Furthermore, we would like to distinguish between two distinct areas inside the hall (e.g. entrance area, service area) where different general rules for customer movement apply. To reflect these changed requirements for the simulation model, we have to reconsider the model (and even architecture) design.

INTRODUCING ACTORS

Facing the problem of detailed modelling of customers' movement and interactions, we still have the option to concentrate the logic of the customers' behaviour into controlling agents (specialized for distinct areas of the hall) and leave the customer entities unintelligent, but this solution is (besides the fact that it does not correspond to the real situation) not flexible, efficient and robust enough. To model such types of systems, ABAsim architecture was missing a reasonable means for modelling of intelligent, initiative entities. Therefore the architecture design was extended by another element called *Actor*. Actors are special type of agents and represent entities (usually humans), which are to some extent autonomous, pro-active, initiative and intelligent. Actors can be seen as a missing element that is placed on a level between agents and entities. Extended ABAsim architecture now contains three distinct categories (levels) of elements with different properties and roles:

- *Agents* are organised into hierarchically structured permanent community, they share the same main goal (system goal, e.g. to serve customers) and cooperate in a common effort to achieve this goal. Agents are not permanently connected with any entity (we can say, agent is "mind without body"). Agents govern temporary groups of actors, and help them to achieve their goals (by providing information, giving advice, etc.).
- *Actors* are agents of lower order representing autonomous, initiative entities, they are able to self control their behaviour in order to reach own goals. Actor is always subordinated to single agent (in this sense, actor is "body with enslaved mind"), though it can migrate to other agent when required. It communicates via messages only with its controlling agent, but can interact with its surroundings (actors, entities). Actors are often mobile; they are created and released during simulation run. Actors receive their local goals from controlling agent, but also have individual interests, which they can follow (as long as these do not interfere with goals assigned by the agent). Unlike agents, actors are not cooperative, they individually follow given goals, but they are not hostile to each other and they respect community rules.
- *Entities* are completely controlled and manipulated by agents (more precisely by agent's assistants). They are not able to control their behaviour (entities can be seen as "mindless bodies").

With the utilisation of augmented ABAsim architecture, we can propose changes to our example model of a service system to respect mentioned new requirements set on the model. Due to the fact, that we now require detailed modelling of customer's behaviour (movement, interactions, etc.), customers have become autonomous elements and actors seems to be the right components to represent the customers. Each customer will be represented by one actor. New proposed model structure is depicted on Figure 2. *Customer Movement Agent* from original model is now replaced with *Customer Agent*, which is responsible for managing customer actors. Unlike *Customer Movement Agent*, *Customer Agent* is not directly executing modelling of customers' movement; this is done by actors, which are representing customers. In the new requirements we have also required that the model respects different behaviour of customer's inside two distinct areas of the service hall (let's call them Area 1 and Area 2). Because customers inside Area 1 behave differently from customers inside Area 2, we will control these areas (in other words, we control actors which are inside these areas) by two agents, responsible for respective areas. All actors which are inside Area 1 are under control of *Area 1 Customer Agent;* if a customer leaves Area 1 and enters Area 2, its control is passed to *Area 2 Customer Agent*. The control of customer actor does not mean the agent is executing any tasks for the actor. Customer actor is autonomous and self responsible for its movement, interactions with surroundings (e.g. avoiding obstacles along the way) and with other customers (e.g. avoiding collisions). Controlling agent can assign goals to actors (e.g. go to the counter) and modify some properties, which define behavioural rules for actors. Actors are then autonomously trying to reach the assigned goals, respecting set rules (e.g. actor is autonomously moving to find the counter, avoiding obstacles and collisions with other actors). After the actor has fulfilled the goal (e.g. it has reached the counter), it notifies its controlling agent by sending a *Done* message.

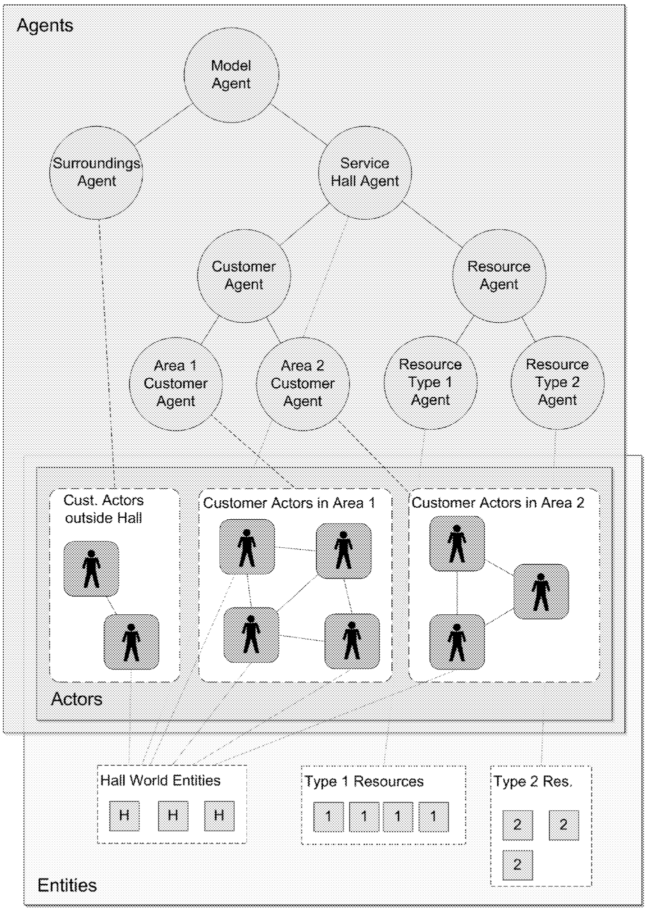


Figure 2: Structure of service system simulation model utilizing actors (enhanced architecture)

INTEGRATION OF ACTORS INTO ABASIM

To enable the symbiosis of agents and actors in a simulation model, few new features were added to the architecture design. In the following text we will try to describe new properties of the ABAsim architecture and integration of actors into simulation models.

Agent-Actor Communication

Communication between agent and actors under its control is executed exclusively via messages. The existing set of message types, which are used for communication among agents and inside agents, was extended by new message types for communication between agent and actor. Actors are able to exchange information with their respective control agent using following types of messages:

- *Done* – message of this type is send by an actor, which successfully reached assigned goal. As a reaction to this message, agent usually assigns a new goal to the actor.
- *Transfer* – actor sends this message to the control agent to indicate, that it is about to leave the domain of current control agent (e.g. actor has reached boundaries of the area, which are controlled by its current controlling agent and wants to move further in the same direction). Agent after receiving this type of message usually identifies agent, which can handle the actor and passes the actor to it (transfers the control of the actor).

To handle the management of actors, agents are using following message types:

- *Goal* – message of this type are used by agents to assign local goals to actors. Actors are obliged to fulfil the goals. However, they can also follow other own interests (take a rest, eat, etc.) as long as these do not interfere with the goal assigned by the agent. After reaching the goal, actors send *Done* message to the agent.
- *Quit* – agents use this message to cancel previously assigned goals to actors. Actors immediately have to stop pursuance of the old goal.
- *Handover* – this message type is used to move actor under control of another agent. Receiving agent becomes new controlling agent of the actor.
- *Entrust* – agent send this message to temporarily move actor under control of another agent. After the actor executes specified tasks, it is expected that the actor will return under control of original agent. Nesting of these temporary handovers is allowed.
- *Return* – is send when agent is returning an actor, which was sent to it using E*ntrust* message.

Agent and Actor Structures

Besides new message type, agent's structure has to be extended by new data fields and methods supporting management of actors, which are controlled by an agent. Each agent holds a list of actors under its control and provides methods for assignment of goals to an actor or group of actors.

In current implementation, actors and agents share the same parent class, i.e. actor has the ability to use the same set of internal components and can exchange messages with its controlling agent. However actors are not able to control other actors – this ability belongs exclusively to agents.

Behaviour of an actor is goal driven. Each actor manages a priority ordered list of goals, which were assigned to it by controlling agent (e.g. actor has to reach the train standing on platform 1) or were put into the list as a result of actor's own interests and desires (e.g. after a specified amount of time an actor can become hungry and would like to visit a restaurant). Since actors can migrate between controlling agents and this migration can be nested, actor has to contain data fields to hold current controlling agent and also a list of past controlling agents.

CONCLUSIONS

Long term experience proved that the ABAsim architecture is flexible and robust base for development of simulation models of large service systems. As a bright example of architecture possibilities and flexibility, simulation tool Villon (formerly called VirtuOS) (Klima et al. 2001, Koenig 2001) can be mentioned. Villon is detailed microscopic simulation model of logistic junction (e.g. railway station, marshalling yard, depot, factory).

Presented changes to the ABAsim architecture were implemented, tested (until now only on a simple demonstration simulation model of passenger's behaviour inside railway transportation terminals) and proven to be functional.

Currently there are plans to utilize the augmented architecture by the design and development of complex simulation model of passenger's behaviour inside terminals (airports, railway stations, etc.) and also to incorporate actors into the design of new modules for Villon simulation software (implementing movement of man-operated vehicles, e.g. fork-lifters in container terminals),

The utilisation of suitable architecture is an important factor in the design and development of a complex simulation model. The presented architecture ABAsim allows the creation of complex simulation models based on agent paradigm. Introducing actors into the set of design elements of the ABAsim architecture broadens the application field of the architecture. Most important properties of the architecture, mainly its ability to support development of complex simulation models, flexibility and reusability of developed models were maintained. The designer of simulation model is free to decide, whether he/she will utilise the new actor paradigm, or will represent simulation entities with "old style" components. The level of autonomy of modelled element should be the main criterion in this decision making process. Intelligent and autonomous elements should be represented using actors.

This work has been supported by the Slovak grant foundation under grant No. 1/1049/04 "Agent Oriented Architecture of Simulation Models of Service Systems" and by the National research program of Czech Republic under project No. MSM 0021627505 "Theory of Transportation Systems".

REFERENCES

Adamko, N., Klima, V., Kavička, A., Lekýr, M. 2004. "Flexible hierarchical architecture of simulation models". In *Proceedings of European simulation and modelling conference*, Eurosis, Paris, 2004, pp.30-34, ISBN 90-77381-14-7.

Henoch, J., Ulrich, H. 2000. "HIDES: Towards an Agent-Based Simulator". In *Proceedings of the Workshop 2000 on Agent Based Simulation*, SCS European Publishing House, 2000.

Klima, V., Kavička, A., Adamko, N. 2001. "Software tool VirtuOS – simulation of railway junction operation". In: *Proceedings of ESM 2001 conference*, Prague, 2001

Koenig, H. 2001. "VirtuOS – Simulieren von Bahnbetrieb". *ETR–Eisenbahntechnische Rundschau*, Hestra-Verlag, Januar/ Februar 2001, pp.44-47 (in German language)

VanPutte, M., Osborn, B., Hiles, J. 2002. "A Composite Agent Architecture for Multi-Agent Simulations". In *Proceedings of the 11th Computer Generated Forces and Behavioral Representation Conference*, Orlando, Florida, 7 - 9 May 2002.

Wijngaards, N., Nieuwenhuis, K., Burghardt, P. 2004. "Actor-Agent Communities in Dynamic Environments". *http://combined.decis.nl/tiki-index.php.*

Wooldridge, M. 1999. *Multiagent systems: A Modern Approach to Distributed AI*, MIT Press, Cambridge, 1999, ISBN 0-262-73131-2

EMOTIONS ON AGENT BASED SIMULATORS FOR GROUP FORMATION

Goreti Marreiros[1], Paulo Novais[2], José Machado[2], Carlos Ramos[1] and José Neves[2]

[1] GECAD – Knowledge Engineering and
Decision Support Group
Institute of Engineering – Polytechnic of
Porto
{goreti, csr}@dei.isep.ipp.pt

[2] Department of Informatics
University of Minho
{pjon, jmac, jneves}@di.uminho.pt

KEYWORDS
Agents, Group Formation and Emotions.

ABSTRACT

Time and space consuming are key factors in a meeting, and therefore must be object of consideration in any process of socialization. So, group decision simulation could be a valuable training tool, through which it will be possible to create and test virtual group decision scenarios. In this work we propose a multi-agent simulator of group decision making that models the participant cortex by considering its emotional states and the exchange of arguments among them.

INTRODUCTION

In daily life we continually make individual decisions, even if we are not conscious of that. The scope of those decisions vary from trivial problems, like 'what clothes should I dress to go to work' to relevant economic and political decisions.

In spite of the great variety of Decision Support Systems, in general they present themselves as simple tools, built according to an user perspective. However, taking decisions in group, rather than individually, may bring some advantages. Aspects like the organizational complexity, the globalization and the internationalization of the markets contribute significantly for the growth of this kind of processes. Taking decisions around table is not an easy task. For instance, many of the decisions of the every day life will acquire a new dimension (e.g. the choice of a place to take vacations, buy a car, hire an employee or select a place to build a new airport). If the group members are dispersed, the need of coordination, informal and formal communication, and information support will increase significantly.

The increase of group decision making processes in organizations contributed to the emergence of Group Decision Support Systems (GDSS). Generically, we may say that GDSS aim to reduce the losses associated to this type of work (e.g. time consuming, high costs, improper use of group dynamics) and to maintain or improve the gains (e.g., groups are better in problems understanding and in flaw detection; participants' different knowledge and processing skills allow for results that could not be achieved individually). The use of GDSS allows for groups to integrate the knowledge of all members into better decision making processes.

Along the last 20 years several GDSS were developed, some dedicated to be used exclusively in decision rooms and others with features to support ubiquitous group decision meetings (Karacapilidis and Papadia 2001; Group Systems/URL).

More recently surged some agent based group decision support systems (Ito and Shintani 1997; Kudenko et al. 2003; Zamfirescu et al. 2001; Payne et al. 2000). On the other hand, simulation proved to be a valuable technique in a range of areas like individual decision making (what if scenarios), e-commerce, crisis situations, traffic simulation, military training, entertainment. Indeed, simulation can be also very useful in the group decision making, once:

- Through it is possible to create virtual group decision scenarios, where the human decision makers can test, for instance, different argumentation strategies and learn from it.
- The training of decision makers is less expensive than the real thing.
- It may be very useful to test "what if scenarios" like, for instance, to test the reaction of to whom was sent an argument with a threat.

The idea of using agent's technology to simulation environments is not new. According to Damasio (Damasio 1994), multi-agent systems offer strong models for representing real-world environments with an appropriate degree of complexity and dynamism.

In previous work, we state that the use of multi-agent systems seems very suitable to simulate the behaviour of groups of people working together and, in particular, to group decision making modelling, once it caters for a broad range of issues, such as individual modeling, flexibility and data distribution (Marreiros et al. 2006).

In classical decision theory proposals are sort by individual decision makers in order to maximize the expected utility. However, if we transpose those choices to quotidian life, it must be taken in consideration that our decisions are influenced by the emotions and moods that one's feeling. The inclusion of affect in individual or group decision processes will allow to explain (simulate) a variety of decisions and observe behaviours, which are difficult to justify under classic decision theory.

There are two different ways to give support to decision makers (Zachary and Ryder 1997). The first one is supporting them in a specific decision situation. And the second one is to give them training facilities in order to acquiring competencies and knowledge that they can use in a real decision meeting.

In this work we propose a multi-agent simulator of group decision making that intents to model the participant cortex considering their feelings, and that allows for the exchange of arguments among them. Decision groups are automatically formed but with the knowledge acquired during the several group decision simulations we intended to model a group formation process.

This paper is organized as follows. Section 2 provides a general approach to role of emotion in decision making processes and presents a brief overview of some of the existent architectures for emotional agents. A model to simulate agent based group decision making is proposed in section 3. We will focus in the emotional component, which is based on the Ortony Clore and Collins (OCC) model (Ortony 2003). Section 4 details some implementation aspects. Finally section 5 presents conclusions and gives some perspectives and ideas for future work.

EMOTION

The terms emotion, mood and affect are many times used indistinctively. Affect is the more general and usually is used to refer to mood and emotion. Emotion is normally referred to as an intense experience, of short duration (second to minutes), with a specific origin, and in general the individual is aware of it. In contrast, moods have a propensity to be less intensive, longer lasting (hours or even days) and remain under unconscious. Moods may be caused by an intense or recurrent emotion, or yet by environmental changes. In what follows the term emotional state will be used to refer to the set of the individual emotions and mood.

Emotion and Decision

Only a few years ago, specialists in decision making started to consider emotion as a factor to be considered in the decision making process. Antonio Damásio (Damasio 1994) proposed a somatic marker hypothesis which describes how emotions are biologically indispensable to decisions. This hypothesis posits that deficits in emotional signal lead to deficient judgment in decision making, especially in the individual and social sphere. According to Damásio, experiments with neurological patients affected by brain damage, shows that the absence of emotion and feelings can break down rationality. In psychological literature several examples could be found on how emotions and moods affects the individual decision making process. For instance, individuals are more predisposed to recall memories that are congruent with their present emotional state. There are also experiences that relate the influence of emotional state in information seeking strategies and decision procedures. The emotional state of an individual has impact in their behaviour, as well as in their interactions with the other group members. The individual emotional state changes with time and is influenced by the emotional state of the remaining members of the group. The process of emotional contagion could be analysed based on the emotions that a group member is feeling or based on the group members mood (Neumann and Strack 2000). A more detailed review of the influence of emotion in group decision making can be found in (Marreiros et al. 2005).

One of the reasons pointed by Rosalind Picard (Picard 2003) to give machines emotional characteristics is the necessity of obtaining a better understanding of the human emotions. As it was seen before, the individual emotional state affects its performance and its relationships inside the group. In this work, it is postulated that the simulation of group decision scenarios will make possible to handle emotions in a way that will allow one to have a better representation and an increasing understanding of the reality.

Architectures for Emotional Agents in MAS

Attending to what has been referred to above, some architectures for emotional agents have been proposed. For instance, Velasquez presents a model called Cathexis (Velasquez and Maes 1997) to simulate emotions, moods and temperaments in a multi-agent system. In his architecture only the basic emotions (i.e., anger, fear, distress / sadness, enjoyment / happiness, disgust, and surprise) are included. In Cathexis, it is presupposed that the simulation of emotional mechanisms implies the interpretation the neurological structures that support emotions. Cathexis follows the somatic marker hypnotises proposed by Damásio. Cathexis was used to implement several synthetic characters like Simón the Toddler (synthetic agent representing a young child) and Virtual Yuppy (a simulated emotional pet robot).

The Flame (Fuzzy logic adaptive model of emotions) emotional model was proposed by El-Nasr (El-Nasr et al. 2000) and is based on fuzzy logic. Flame is composed by three models: emotional, decision making and learning. The emotional model is mainly based on the OCC model. Flame architecture is designed for a single agent, and does not incorporate functionalities related to group behaviour.

Urban and Schmidt propose the PECS (Physics, Emotion, Cognition, and Social Status) reference model (Schmidt 2002; Urban 2000). PECS is an architecture for multi-agent systems, which aims modelling and simulating the human behaviour. PECS agents contain information that falls into four categories: physical (the agent's physical condition), emotional (agents feelings), cognitive (agents plans, model of the self and model of the environment) and social status (relations in the community of agents). A simulation model named Adam was developed to test and demonstrate the PECS architectures capabilities.

Salt&Peper architecture was proposed by Luís Botelho and Hélder Coelho (Botelho and Coelho 2001). Salt&Peper is architecture for autonomous agents that aims to implement mechanisms to allow artificial agents being as successful as natural agents. The roots of this architecture are in neuroscience and cognitive science; the authors boost the adaptive role of emotions. Generically we may say that the architecture aims to develop control mechanisms to artificial agents that are emotional based. The Safira project uses the Salt&Peper architecture for the implementation of its agents (Paiva et al. 2001).

Hyungil Ahn and Rosalind Picard proposed a computational framework of affective-cognitive learning and decision making for affective agents. This framework is inspired by human learning, neuroscience and psychology (Ahn and Picard 2006).

SIMULATOR DESCRIPTIONS

In previous work, it was identified the main agents involved in the simulation of a group decision (Marreiros et al. 2006), namely: Participant Agents, the Facilitator Agent, the Register Agent, the Voting Agent and the Information Agent.

In the remain of this section, we will focus on the architecture of participant agents, due to their main role in group decision making, and in particular on the Emotional module. At this moment, only the participant agents have emotional characteristics, however is possible to extend this characteristics to the facilitator agent (the responsible for the simulation).

Participant Agent Architecture

In figure 1 it is represented the architecture of participant agents. This architecture comprises the knowledge layer, the reasoning layer and the communication layer.

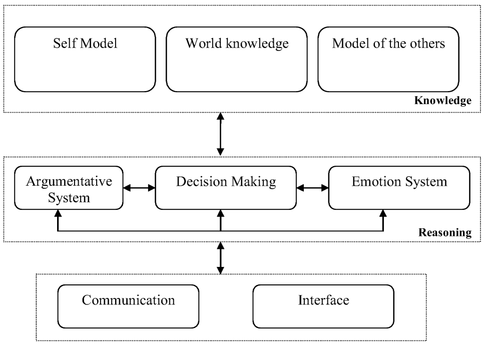


Figure 1: Participant Agent Architecture

In the knowledge layer, the agent has information about the environment where it is situated, about the profile of the other participant agents, in terms of its own preferences and goals. The information in the knowledge layer is dotted of uncertainty, evolving according to the agent interaction with its peers.

The communication layer is responsible for the communication among agents and the user interface.

The reasoning layer contains three major modules:

- The argumentative system – that is responsible by the arguments generation. This component will generate explanatory and persuasive arguments, which are related to the agent emotional state and about its thinking on its peers (Analide and Neves 2002).
- The decision making module – it will support agents in the choice of the preferred alternatives and will classify all the set of alternatives into classes, namely preferred, indifferent and inadmissible.
- The emotional system – it will generate emotions and moods, affecting the choice of the arguments to be sent to the other team members, the evaluation of the received arguments and the outcome.

Emotional Module

The emotions that will be simulated in our system are those identified in the reviewed version of the OCC (Ortony 2003) model, namely, joy, hope, relief, pride, gratitude, like, distress, fear, disappointment remorse, anger and dislike.

An emotion in our system is characterized by the following properties: if it is positive or negative, moment in time when it was initiated, identification of the agent or event that cause the emotion and emotion intensity.

The user will setup a set of rules to configure the emotion generation. The system is prepared to allow the configuration of all the set considered in the OCC model, but the user may just opt to configure a subset of that.

In Figure 2 it is possible to visualize the main components of the emotional system.

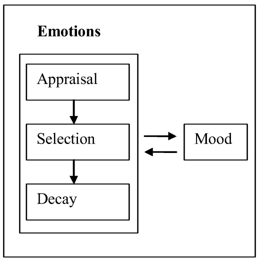


Figure 2: Emotional Module

The emotional module is composed by three main components: appraisal, selection and decay. The agent mood is determinate based on the emotions felted.

Appraisal

The appraisal mechanism is based on OCC model, where the user defines the conditions for the emotion activation. An example may be:

$$Hope(AgPi,X)\text{:-}Goal(AgP_i,X),\quad Request\ (AgP_j,X).$$

In the previous example, the emotion Hope is appraised if Agent AgP_i has the goal (X) and asks to agent AgP_j to perform the goal X. A weight, in the interval [0,1], is settled for each condition of the emotion generation. The emotion intensity is computed according the conditions weight. A particular emotion depends on the intensity of the others emotions.

Selection

All the emotions defined in the simulator have a threshold activation that can be influenced by the agent mood. The activation threshold is a value between 0 and 1. This component selects the dominant emotion.

$AgP_{i,Emo,t}$ is the set of all the emotions generated by the agent AgP_i and the respective intensities and activation thresholds.

$AgP_{i,Emo,t}=\{(Emo_1,Int_1,Act_1),\dots(Emo_n,Int_n,Act_{n)}\}$

The selected emotion in instant t, $AgPi_{ActEmo,t;}$ will have a higher differential between the intensity and the activation.

Decay

Emotions have a short duration, but are not instantaneous (they have a period of decay). There are several proposals for this calculation. In our model, it is considered three possibilities, namely, linear, exponential and variant.

The decay rate may be the same for positive and negative emotions. It is also possible to settle different rates for positive and negative emotions, in that case the user should choice the variant decay rate.

Mood

The agent mood is calculated using the felt of emotion in the past and what the agent thinks about the moods of the remaining participants. In our approach, the process of mood contagion is the only one to be considered. The process of emotion contagion is handled. We consider three stages for mood, namely, positive, negative and neutral. The mood of a specific participant is determined according to the following:

$$K^{+} = \sum_{i=t-n}^{t-1} I_i^{+} \, , \, K^{-} = \sum_{i=t-n}^{t-1} I_i^{-}$$

K^+ and K^- are the sum of the positive/negative emotions felt in the last n periods, and n can be parameterized by user. Only emotion values above the threshold activation are considered.

$$\begin{cases} \mathit{if}\ K^{+} \geq K^{-} + l, \ \mathit{then\, positive\, mood} \\ \mathit{if}\ K^{-} \geq K^{+} + l, \ \mathit{then\, negative\, mood} \\ \mathit{if}\ \left| K^{+} - K^{-} \right| < l, \ \mathit{then\, neutral\, mood} \end{cases}$$

The value of l varies according to the mood of the remaining group members.

$$\begin{cases} l = 0.10, \ \mathit{if\ group\, mood\, is\, positive} \\ l = 0.05, \ \mathit{if\ group\, mood\, is\, neutral} \\ l = 0.01, \ \mathit{if\ group\, mood\, is\, negative} \end{cases}$$

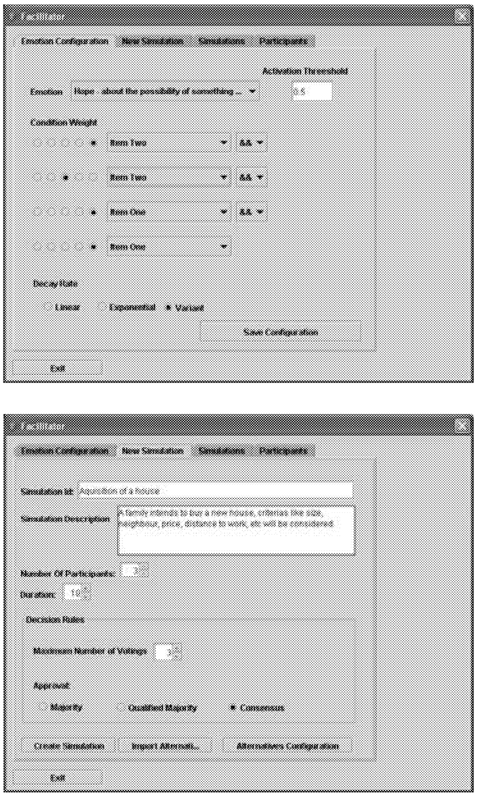
Each participant agent has a model of the other agents, in particular, it has information about the other agent's mood. This model considered incomplete information handling and the existence of explicit negation, following the approach described in (Analide and Neves 2002). Some of the properties that characterize the agent model are gratitude debts, benevolence, credibility, (un)preferred arguments, and reputation (Andrade et al. 2005). Although, the emotional component is based on the OCC model. The inclusion of mood can surpasses one of the major critics that usually is pointed to this model, the fact that OCC model does not handle the treatment of past interactions and past emotions

IMPLEMENTATION

A prototype of the multi-agent model proposed in the previous section is being developed in order to validate the model. In this section, we present some details of our implementation.

The prototype is being developed in Open Agent Architecture (OAA) (OAA -URL)], Java and Prolog. The participant agents are being developed in Prolog, while the other agents that compose the proposed model are developed in Java. The implementation of the AgPs in Prolog is related to the existence of incomplete and negative information in the knowledge base of each AgP, and on the necessity of measuring the quality of that information. OAA has an Interagent Communication Language (ICL) that is shared by all agents independently of the language in which they are programmed or the operating system of the machine where the agents reside. The ICL language is close to KQML. OAA imposes a common protocol for agents entering and registering at the group decision making simulator.

In Figure 3, it is possible to see the emotional configuration process. In this particular case the user is configuring the emotion Hope that has an activation threshold of 0.5 and the decay rate used is variable. Figure 3 presents a configuration of a simulation; in this case the goal is to simulate the acquisition of a house by a family.



Emotion: Hope - about the possibility of something ...

Activation Threshold: 0.5

Condition Weight

● Item Two &&

● Item Two &&

● Item One &&

● Item One

Decay Rate: ○ Linear ○ Exponential ● Variant

Simulation Id: Acquisition of a house

Simulation Description: A family intends to buy a new house, criterias like size, neighbour, price, distance to work, etc will be considered.

Number Of Participants: 3

Duration: 10

Decision Rules

Maximum Number of Votings: 3

Approval: ○ Majority ○ Qualified Majority ● Consensus

Figure 3: The simulator parametrics

Figure 4 shows the agents that exist at a particular moment in the simulator: 10 participant agents, the facilitator agent, the voting agent, the clock agent (OAA is not vocalized for simulation, therefore it was necessary to introduce a clock agent to control the simulation) and the application agent (responsible by the communication between the community of agents and the simulator interface).

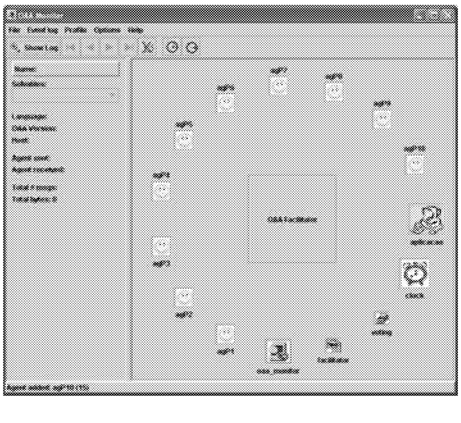


Figure 4: Community of agents

CONCLUSION

We propose an agent based group decision simulator, which aims to simulate the behaviour of persons involved in group decision. In this simulator, each group member is represented by a separate agent, which facilitates the simulation of entities with different behavioural characteristics. The inclusion of an emotional module will allow for its users to obtain a better representation of the reality. The simulator is flexible, once it is easy to add or remove a participant from the scenario during a simulation. This work is focused on the emotional system. However another important component of the participant agent architecture is the argumentation system that has been already approached in (Marreiros et al. 2006).

Futures developments of this model will include factors like credibility, reputation and the member hierarchy inside the organization.

ACKNOWLEDGEMENTS

This work is supported by ArgEmotionAgents project (POSI / EIA / 56259 / 2004), a project of FCT (Science & Technology Foundation – Portugal).

REFERENCES

Ahn, H. and R. W. Picard. 2006. "Affective Cognitive Learning and Decision Making: The Role of Emotions," The 18th European Meeting on Cybernetics and Systems Research, Austria.

Analide, C. and J. Neves. 2002. "Antropopatia em Entidades Virtuais". I Workshop de Teses e Dissertações em Inteligência Artificial, Brazil (in Portuguese).

Andrade F.; J. Neves; P. Novais; J. Machado; and A. Abelha. 2005. "Legal Security and Credibility in Agent Based Virtual Enterprises", in Collaborative Networks and Their Breeding Environments, Camarinha-Matos L. Afsarmanesh H., Ortiz A., (Eds), Springer-Verlag, ISBN 0-387-28259-9, pp 501-512.

Botelho, L. and H. Coelho. 2001. "Machinery for artificial emotions", Cybernetics and Systems Vol.32 No 5 pp. 465-506.

Damasio, A. 1994. "Descartes' Error: Emotion, Reason and the Human Brain", Picador.

Davidsson, P. 2001. "Multi agent based simulation: beyond social simulation", Proceedings of the Second international Workshop on Multi-Agent Based Simulation S. Moss and P. Davidsson, Eds. Springer-Verlag New York, pp. 97-107.

El-Nasr, M; J. Yen; and T.R. Ioerger. 2000. "FLAME - Fuzzy Logic Adaptive Model of Emotions". Autonomous Agents and Multi-agent systems, Vol.3 pp. 217-257.

GroupSystems –URL: www.groupsystems.com

Ito, T. and T. Shintani. 1997. "Persuasion among agents: An approach to implementing a group Decision Support System based on multi-agent negotiation", Proceedings of the 5th International joint Conference on Artificial Intelligence.

Karacapilidis, N. and D. Papadias. 2001. "Computer supported argumentation and collaborative decision making: The Hermes system", Information Systems, Vol. 26 No. 4 pp. 259-277.

Kudenko, D.; M. Bauer; and D. Dengler. 2003. "Group decision making through mediated discussions". Proceedings of the 10th International conference on user modelling.

Marreiros, G.; P. Novais; J. Machado; C. Ramos; and J. Neves. 2006. "A formal approach to argumentation in group decision scenarios". Knowledge and Decision Tecnhnologies, Vale Z., Ramos C. and Faria Luiz (Eds), ISBN 972-8688-39-3, pp 135-141.

Marreiros, G.; C. Ramos; and J. Neves. 2005. "Emotion and Group Decision Making in Artificial Intelligence", Cognitive, Emotive and Ethical Aspects of Decision-Making in Humans and in AI vol IV. Ed. Iva Smit; Wendell Wallach; George Lasker, Published By The IIASS, ISBN 1-894613-86-4, pp 41-46.

Neumann, R. and F. Strack. 2000. "Mood contagion: The automatic transfer of mood between persons", Journal of Personality and Social Psychology, Vol. 79 pp 211-223.

OAA-URL: www.ai.sri.com/~oaa/.

Ortony, A. 2003. "On making believable emotional agents believable", In R. P. Trapple, P. (Ed.), Emotions in humans and artefacts. Cambridge: MIT Press.

Paiva, A.; E. André; Y. Arafa; L. Botelho; M. Costa; P. Figueiredo; P. Gebhard; K. Höök; A. Mamdani; C. Martinho; D. Mourão; P. Petta; P. Sengers; and M. Vala. 2001. "SAFIRA- Supporting Affective Interactions in Real-time Applications". CAST - Living in mixed realities, Special Issue of netzpannung.org/journal.

Payne, T.; T.L. Lenox, S. Hahn; M. Lewis; and K. Sycara. 2000. "Agent-Based Team Aiding in a Time Critical Task", Proceedings of the 33rd Annual Hawaii International Conference on System Sciences, IEEE Press, Los Alamitos.

Picard, R. 2003. "What does it mean for a computer to have emotions?"; In Trappl, R.; Petta, P.; and Payr, S. (eds) Emotions in Human and Artefacts.

Schmidt, B. 2002. "How to give Agents a Personality", Proceedings of the 3rd Workshop on Agent-Based Simulation, Germany. SCS-Europe, pp. 13-17.

Urban, C. 2000. "PECS – A Reference Model for the Simulation of Multi-Agent Systems", in: Suleiman, R., Troitzsch, K. G., Gilbert, G. N. (eds), Tools and Techniques for Social Science Simulation, Physica Verlag, Heidelberg.

Velasquez J. D. and P. Maes. 1997. "Cathexis: A Computational Model of Emotions," Proceedings of the First International Conference on Autonomous Agents, pp. 518-519.

Zachary, W.; J. Ryder. 1997. Decision Support Systems: Integrating Decision Aiding and Decision Training. In M. G. Helander, Landauer, T.K., and Prabhu, P. (Ed.), Handbook Of Human-Computer Interaction, 2nd Edition (pp. 1235-1258). The Netherlands: Elsevier Science.

Zamfirescu, C.B.; C. Candea; and S. Luca. 2001. "On Integrating Agents into GDSS", Proceedings of the 9th IFAC/IFORS/IMACS/IFIP Conference on Large Scale Systems, Bucharest, pp. 231-236.

TIME AND SPACE MANAGEMENT IN CROWD SIMULATIONS

Benoit Lacroix
Philippe Mathieu
Sébastien Picault

LIFL CNRS UMR 8022
Université des Sciences et Technologies de Lille
Cité Scientifique bat. M3
F-59655 Villeneuve d'Ascq Cedex
E-mail: {lacroixb,mathieu,picault}@lifl.fr

KEYWORDS

Multi-Agent simulation, Software Engineering, Emergence, Auto-Organization

ABSTRACT

In this paper, we present a contribution to time and space management in reactive simulations, especially for crowd simulation in an urban or closed environment. We are not interested in mere individual scheduling, but rather in the link between global rythm issues and individual behavior. Our aim is to simulate real moving behaviours in constrained environments, taking into account the link between the macroscopic predictions of the simulation and experimental data. The simulation must deal with collision problems, rythms variations or multiple moving models. We argue here that a synthesis between temporal and spatial features is needed. This work relies upon our MISC model which allows to simulate 2D-moves with different rythms according to population classes, and gives a statistical control on the life cycle of the individuals.

INTRODUCTION

Our objective is to simulate pedestrians motion through complex environments. Space is to be considered on the level of modelisation of pedestrians and their environment, like with that of the motion models used. A temporal dimension has to be added: the simulation must proceed in a "simulated" time. We have to make sure that a transformation exists which makes it possible to convert this "simulated" time to an "exact" one for which simulation is constantly in phase with the real situation. Finally the validation of the simulation is also one of the most important concerns. It imposes on the time problematics, and implies the require to be able to confront the obtained data with reality, and to reproduce real behaviors (fig. 1).

Only the issue of exact simulations, where it is necessary to be able to validate the results by an adequate step rather than providing a mere realistic appearance, is considered here. Two forms of prediction can then be considered: at the macroscopic level first, with flows at entrances or exits and distributions in space and time; and second at the microscopic level, by analyzing individual behaviors and comparing them with data available from psychology and sociology.

This work led to the development of the MISC platform (for "Modeling of Individuals in Spatialy Constrained environment"), which allows the simulation of reactive entities. (This work is supported by the French CPER TAC of Region Nord-Pas-de-Calais and by European Funds from FEDER.) The agents which simulate moving persons employ algorithms of collisions and adjust their movements in an autonomous way according to their particular characteristics and their limited perceptions. The model used is based on the concepts of blurred path following and sources of agents, those being able to be parameterized to generate agents of a given family at specified intervals. Being able to take into account all the possible situation, from point-to-point displacement to queuing at a ticket distributor, is also a problem which has been studied, through the use of multiple models of placement/displacement. Global measurements of spatial and temporal aspects of the simulation are provided, in order to compare the predictions of the simulator with the real data.

We will first present the state of the art in the field of pedestrian studies and simulations of displacement, and then describe the main difficulties involved: it is mainly about how to take time and space into account, and to be able to validate the results of simulations. We also present the tool which results from the integration of this temporal model to models of realistic displacements in this context. Finally our proposals for the resolution of these problems are presented, before indicating the prospects for our work.

RELATED WORK

The modeling of the displacement of human individuals is an active subject of research which was approached along various models, roughly divided into three categories. The first one compares the individuals to particles to carry out analogies with fluid or gas dynamics. However these models, by

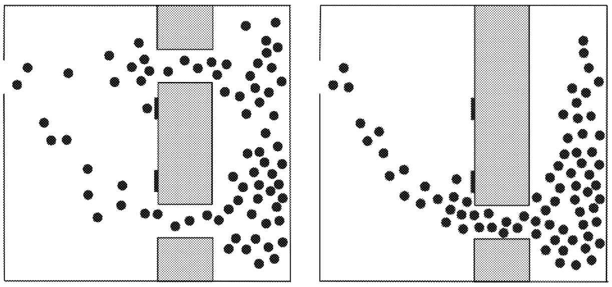


Figure 1: An example of the kind of problems we aim at resolve : is the organization of this train station more efficient with one or two exits, and what is the influence of the placement of the exit(s) ?

treating individuals in a uniform way, lead sometimes to predictions which do not correspond to reality (Kerridge et al. 2001). The second approach consists in discretizing space in cells, for example by representing the agents and their environment by a cellular automata. The displacement of the individuals and the physical occupation of space are then determined by the rules of the automat. This approach presents however two main difficulties (Still 2000), such as a limitation of the capacity of space description, and difficulties in implementation and interpretation of results. Finally the last category gathers the models based on a continuous environment and privileging the emergence of behaviors from individual rules, like the Magnetic Force Model (Okazaki 1979) or the Social Force Model (Helbing and Molnar 1995), based on three forces (the motivation of achieving goals, interactions with other individuals and environment, and attractive effects of the environment). Nevertheless, the fact that the model is not based on real data makes it difficult to parameterize and validate.

Many platforms apply these models. We will describe here three ones more in detail, in order to have an outline of their possibilities. First, K. Teknomo tries to model the behavior of pedestrians in an as exact as possible way, while being particularly interested in the possibilities of validation (Teknomo 2002). The model used is based on the Social Force Model. Each pedestrian is subjected to three forces: one of attraction (making the agents advance towards their destination), and two of repulsion (for target anticipation and collision avoidance). This approach has the advantage of being based on measurable physical data, by construction of the used forces. However this model does not incorporate in its current version the management of other obstacles than the agents themselves, and the interactions with the environment are then limited.

G. Keith Still, in its study of crowd dynamics, has for objective to obtain an emergent simulation from the simplest possible rules (Still 2000). The system is based on four behavioral rules, and one displacement rule. Each agent must try to reach its objective (fixed, or imposed by a given situation), while trying to get to its maximum speed. It must also maintain a minimal distance with the other objects of its environment, while having a certain reaction time to the external events. The rule of displacement is based on the principle of the least effort, making it possible for the agent to choose the shortest way in term of displacement time. This model makes it possible to reproduce certain charasteristical behaviors of the pedestrians, like their capacity to use short cuts, or the formation of lines in bidirectional flows. It remains however difficult to gauge compared to the real data.

C. Reynolds uses a different point of view: he does not care about obtaining exact results, but desires a probable and effective simulation (Reynolds 1999). His work, inspired by robotics, intends rather for the design of games or animations, but is nevertheless interesting, since it aims at reproducing realistic behaviors. Each agent uses elementary behaviors, exploited according to the circumstances, such as the *arrival* (consisting in slowing down near the objective), the *obstacle avoidance* (adjustment of the vision distance according to the speed so as to anticipate the skirting of the obstacles), or the *path following* (the agent follows a preset way made of broken line, adapting his speed in order not to move too far away from it). The association of these various elementary behaviors makes it possible to obtain complex and realistic simulations.

TIME AND SPACE IN AGENTS SIMULATIONS

Pedestrian studies involves complex phenomena, and the emergence of realistic behaviors like the possibilities of validation are related to the resolution of fundamental questions, which have to be answered during the conception. We will first consider the individual characteristics which seem essential to us.

Taking individual features into account

A first parameter is the space occupied by a person. It expresses its morphological characteristics, and can depend on various parameters An other important factor likely to influence the behavior of the individuals is the concept of personal space. Each person defines around itself a private zone, which it tries to preserve of the influence of the others (Sommer 1969) or to regain if it is reduced. People also have an individual velocity, regarded as a preferred speed. This speed will also be function of various factors.

As for personal space, the characteristics of the person are important, but also depend on external parameters. Last, a person perceives its environment. These perceptions will determine the possibility to anticipate and to react to the interactions with other agents and obstacles. The visual field for example has interesting characteristics, which can be taken into account. The manner and the capacity to perceive the environment also depend on factors like size or density: small or tall people, dense or sparser crowd will be as many elements which will influence perceptions.

Types of displacements

As we have seen, various models of displacements were developed. However their characteristics are not perfectly appropriate for all situations, and it is difficult to conceive a model which would take all of them into account. For example, to take a train an agent must first go to the train, then wait on the platform, next go up on board and finally find a place. During the phases of displacement a traditional model as those described above could be appropriate to go to a precise place under the constraints of the environment. But during the waiting phase, rather than staying at a precise point the agent is likely to hover around in a small zone while respecting constraints such as the density of the crowd. It can thus be more accurate to change behavior, and thus of type of displacement during the time, rather than to seek to design a global and consequently complex model, to bring an adequate solution to all possible situations.

Complex behaviors to reproduce

In the reality individuals often adopt complex, adaptive behaviors. While moving, an agent sometimes has to move backward to leave the way to another one. When advancing towards a precise goal, such a behavior is a difficult problem, even in the case of co-operative robots where the space of evolution is bigger (Lucidarme et al. 2002). It is the same for behaviors in waiting queues or areas. Moreover those behaviors depend on the culture or the social standards.

In addition some emergent phenomena can be observed, such as the formation of lines in multidirectional pedestrians flows or flocking/herding effects caused by the sociability of individuals. It is also important to offer to the agents the capacity to anticipate their actions, in order to observe natural behaviors.

Various manners to take time into account

Time can be taken into account through various manners, according to the required objective. In the first approach, i.e. real time simulation, the outputs of the simulation correspond at any time at the precise current state of simulation. In a second approach where time could be qualified as an “exact” one, the objective is to obtain an output which is constantly in phase with what would occur in the real world. The observation of reality and that of simulation must then have an exact correspondence during the time. Finally a last approach is the “simulated” time, where a mathematical transformation to convert simulated time to exact time has to exist.

This last approach makes it possible to accelerate or slow down simulations, the maximum speed corresponding to the speed limit of calculation of the platform used. Each step of time does not have the obligation to use a constant computing time, as long as the correspondence with the “exact” time is maintained.

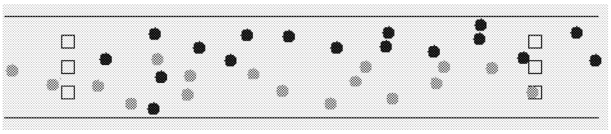


Figure 2: The MISC platform running a simulation of bidirectionnal flows in a corridor. We can observe the formation of lines and that fast people pass the others.

The issue of the validation

One of the most important problem in this type of simulation is the issue of the validation. Validation can be obtained at two different levels: macroscopic (input/output flows) and microscopic (analyze of the individual behaviors). To be able to extract these two kinds of data during the course of the simulation is then very important, in order to be able to check their exactitude.

A crowd can be characterized by various parameters: average distributions, values, flows, densities. These macroscopic characteristics are very often the only ones which can allow a validation from real data, being the only available ones. The concept of level of service (Fruin 1971) raises on the establishment of relations between flow, velocity and density. To reproduce these relations ensure a certain validity of the results of simulation. However, as Keith Still underlines it (Still 2000), the sociological studies on this subject generally correspond to a particular context. The confrontation of their results with those of simulations is then particularly difficult, because of the complexity of transposing the context.

The validation from a microscopic point of view is less clearly defined. A regularly proposed aspect relates to the capacity of the model to reproduce auto-organization phenomena observed in reality, as those mentioned above (fig. 2). The interest of a model of displacement based on real and measurable physical data is also here clearly visible. In addition to allowing easier calibration, the input parameters can be directly fixed by observations in the real world.

OUR PROPOSAL: THE MISC PLATFORM

Each one of the existing platforms make it possible to answer some of the problematics raised above, but none is today able to solve the totality of them. The approach we implemented with the development of the MISC platform aims to bring a solution to these various problems. We will see now an outline of the answers we gave them.

The agents and their environment

First of all can the pedestrians have various profiles. These profiles determine their physical characteristics (size, preferred velocity…) and their perception capacities. These various elements can be allotted grouped or individually managed in order to allow a great flexibility in simulations.

Their behavior, within the determination of the objectives that they will be seen assigned, will be discussed further below.

The generation and the disappearance of the agents are managed by elements called sources and wells. Sources generate the agents using temporal criteria, in order to integrate flow data. The wells constitute the final goals of the agents, and remove them from the simulation.

Environment, in addition to the static elements constituting the architecture of the places (entries and exits, fixed obstacles...), can include elements likely to interact directly with the agents. The first type of elements gathers those able to divert or capture the attention of the agents. They can lead them to introduce additional intermediate goals into their objectives (stands, shops...), or to allow the realization of goals initially present (ATM, vending machines...). The activity near these elements is modified and thus has an incidence on flows and the spatial distribution of the agents.

Another type corresponds to purely informative elements, which modify the knowledge of the agents and influence their behavior, such as traffic signs. Some agents, though they are nonfamiliar with the environment, can use these informations to achieve their goals. On the contrary an agent knowing the environment will not take them into account.

The behavior

The behavior of the agents uses a three-level logic: a strategic level, a tactical level and a level of local resolution.

The strategic level corresponds to the determination of the various goals the agent will have to reach during its life cycle. For example an agent using the subway can thus have to reach the objectives “to take the subway”, “to stay in the subway during three stations” and then “to go out and to leave the arrival station”.

The tactical level determines the local objectives that the agents have to achieve to reach their strategic objectives, while taking into account their individual characteristics. A strategic objective “to take the subway” can thus be split in various manners. A tourist, nonfamiliar with the environment, must determine his access path to the platform before being able to go there, or at least while going there. On the contrary a person knowing the station can immediately determine its way, because of its familiarity with the organization of the station. Another example is the roundabout, where all the agents have to adopt the same direction of circulation (fig. 3). The simple calculation of the shortest way does not make it possible to respect the circulation rules.

Finally the last level corresponds to the choice of local resolution. It can for example be the calculation of the next point where the agent will be, by applying the method of displacement determined at the tactical level, or the wait for an additional step of time if the agent is in a queue.

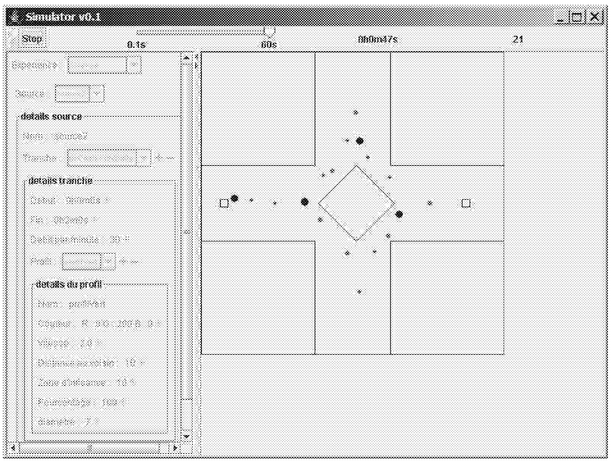


Figure 3: The simulator with various types of agent, having varied characteristics (size, speed...), which moves around a central rhombus representing a roundabout. It is not possible to circumvent this kind of obstacle and respect the circulation rules only by using flocking or shorter way.

This decomposition includes feedback and backtrack mechanisms to take into account local constraints when achieving higher goals.

Displacement

As for the models of displacement to be used, two particularly interesting approaches can be combined: the blurred path following, and a variant of the Social Force Model. The blurred path following takes as a starting point the method implemented by Craig Reynolds for path following. After having determined its next point of destination, the agent tries to join up with it while following a straight line, and respecting various constraints. It must locally solve the collisions constraints with the other agents and the static elements of the environment, and remain as close as possible to the direct trajectory. One of the main advantages of Social Force Model is the capacity of anticipation it offers to the agents. The use of forces inspired by it for the resolution of the local constraints would then make it possible to integrate it into our model.

Simulated time

The chosen approach for the management of time is the simulated time. Our agents do not necessarily have to be conscious of time, as long as they can be precisely managed by the simulation. However this flexibility is obtained only by taking care of various points.

First of all it is essential to be able to preserve a total control of the scheduling of the various tasks. An intuitive approach which would be to try an implementation in traditional multitask would likely lead to failure. A specific scheduler must be set up.

In addition to the rhythm given to the agents, the temporal concept is integrated in the notion of source. Sources are the entities that govern the creation of the agents, according to various parameters. The objective is to offer an operation by temporal phase. This way macroscopic data like input flows can be integrated and translated at the macroscopic level by the corresponding agents.

A big difficulty encountered in this type of simulation is the management of the waiting points, like lineups or waiting zones. Without an adequate model, instabilities or blockings can occur in special points of activities. A switch between different modes which modifies the behavioral rules between the phases of waiting and activity is then necessary. In order to be able to validate in experiments the effects of these activities, it is necessary to attempt to reproduce their rhythmic characteristics (flows of entries, flows of exits, etc), rather than the detail of the individual behaviors that are not relevant for the rest of the environment in the level of local resolution of displacements. From this point of view they will have space and behavioral characteristics which will evolve according to the number and the type of agents on standby.

The validation of the results

In terms of validation, we will first carry out a macroscopic validation using flows in clearly identified points of passage, spatial distributions of the agents, densities, etc. The real data are acquired by counting (for flows) or by video (for the distributions) and are confronted with the results of simulation. Those can in this case be calculated offline (without visualization of the individual behaviors). In addition, we can also carry out a microscopic validation of the behavior of the individuals. For example, the detail of the displacement of the agents can be observed and then compared with the data of the literature; it is also possible to study the "emergent" group behaviors such as circulation in file or congestion.

The problem of the scheduling of the tasks has also to be considered, in order to determine a policy of activation of the agents within the simulations. Indeed the choice of this policy is not neutral for the obtained results: an inequity in the treatment of the agents could significantly influence them. A satisfactory solution is to carry out a random draw between the agents at each step of time in order to determine scheduling, a solution which can still be improved by entangling the various behaviors. This last solution implies however that we have to set up a complex scheduler, after having carried out a classification making it possible to preserve the integrity of the simulation.

CONCLUSION AND FUTURE WORK

In this paper we proposed an approach mixing the problematics of displacement with a proposal for the taking into account of time in agents centered simulations. The objective here is to offer a model and tools adapted to crowd simulations with temporal constraints. In addition to the problems of collisions, considerably discussed in the literature, one of the difficult problems is to make it possible for the agents to change their system of displacement during their move towards the objective, and to have a global and effective system of management of the temporal constraints. The approach presented here and implemented in the MISC platform is based on the concept of sources of agents giving rythm to their life cycle from temporal constraints. These sources assign objectives to the agents, objectives that will be solved by their behavioral logic. Their displacements follow blurred paths, which are marked with stop points in queues. An agent is thus able to make half of his way, to wait a few minutes and then to start again to carry on its way. We use a method for the validation of simulation with measurement of flows of exit of the agents. The MISC platform, already operational, enables us to carry out experiments on varied levels and to advance towards our next stage: validation based on real data.

REFERENCES

Fruin J.J., 1971. *Pedestrian Planning and Design*. Metropolitan Association of Urban Designers and Environmental Planners, New York.

Helbing D. and Molnar P., 1995. *Social Force Model for Pedestrian Dynamics*. *Physical Review E*, 51, 4282–4286.

Kerridge J.; Hine J.; and Wigan M., 2001. *Agent-based modelling of pedestrian movements: The questions that need to be asked and answered. Environment and Planning B: Planning and Design*, 28, 327–341.

Lucidarme P.; Simonin O.; and Ligeois A., 2002. *Implementation and Evaluation of a Satisfaction/Altruism Based Architecture for Multi-Robot Systems*. In *Proc. of IEEE Conference on Robotics and Automation (ICRA'02)*. 1007–1012.

Okazaki S., 1979. *Study of Pedestrian Movement in Architectural Space, Part 1: Pedestrian Movement by the Application on of Magnetic Models*. *Trans of AIJ*, , no. 283, 111–119.

Reynolds C.W., 1999. *Steering Behaviors For Autonomous Characters*. In *Game Developers Conference 1999*. Miller Freeman Game Group, San Francisco, California, 763–782.

Sommer R., 1969. *Personal Space: The Behavioral Basis of Design*. Prentice Hall.

Still G.K., 2000. *Crowd Dynamics*. Ph.D. thesis, University of Warwick.

Teknomo K., 2002. *Microscopic Pedestrian Flow Characteristics: Development of an Image Processing Data Collection and Simulation Model*. Ph.D. thesis, Tohoku University Japan, Sendai.

PROCESS SIMULATION WITH AGENTS

AGENT-BASED MODELING OF PROCESSES AND SCENARIOS WITH HIGH-LEVEL PETRI NETS

Timo Steffens, Thomas Zöller, Philipp Hügelmeyer
Fraunhofer IAIS
Schloss Birlinghoven, 53754 Sankt Augustin, Germany
email: timo.steffens@iais.fraunhofer.de

KEYWORDS

Agent-based simulation, high-level petri nets

ABSTRACT

This paper introduces an approach to simulation which combines agent-based modelling and high-level petri-nets. The first main idea is to use concepts of high-level petri nets in order to support different abstraction levels and to effectively handle parallelisms. The second main idea is to model all entities as agents, including environment objects, IT systems or classical agents. That is, all events and effects are handled as actions of agents. Third, agents are grouped into clans in order to facilitate modelling and increase performance.

INTRODUCTION

Most real-world processes are complex and consist of several subprocesses running in parallel. In order to analyze or optimize such processes, sophisticated methods for modelling and simulation are necessary. Modelling a process serves as documenting and understanding the essential dynamics of the system on the one hand, and as prerequisite for simulation on the other hand. This paper presents an approach for modelling that combines the concepts of multi-agent systems (Weiss 2000) with high-level petri nets (Jensen 1992). To this end, we introduce the language LAMPS (Language for Agent-Based Modelling of Processes and Scenarios) and the simulation system called LAMPSSys.

LAMPS can be used to document and analyze processes and to directly implement corresponding simulations. LAMPS is designed to have a broad application scope, so that a wide range of processes can be modelled, including business processes, warfare scenarios (Huegelmeyer et al. 2006), critical infrastructures (Flentge and Steffens 2006), or physical simulations. This wide applicability is due to LAMPS' generic approach which uses agents to represent all active entities. All entities in the system, including environment objects, IT-systems, or classic agents, can be modelled in the same way as agents. All effects and events are modelled as actions of agents.

LAMPSSys is the simulation environment that can run agents whose behavior is specified in LAMPS. The system handles time, load and communication between the agents. The system is designed to be extensible and generic in order to be compatible to standard tools.

This paper is structured as follows. In the next section we discuss related work. Following this, we describe the language LAMPS. Afterwards, the agent model that is the core of LAMPSSys is introduced. A section about details of the LAMPSSys architecture follows. Then, applications of LAMPSSys are outlined, and finally the last section concludes.

RELATED WORK

While languages for the specification of simulation models abound, LAMPS is to our knowledge the only one that supports both the description of scenarios and the executable specification of agent behavior for compound agent groups down to individual agents. Existing simulation programming environments are usually extensions of programming languages such as C/C++ (e.g. Maisie (Bagrodia and Liao 1994), the SPaDES environment (Teo et al. 1998)) or Java (e.g. SILK (Healy and Kilgore 1997), the SSJ package (L'Ecuyer and Buist 2005)). In contrast, LAMPS is based on high-level Petri nets (Jensen 1992). Thus, LAMPS inherently supports parallel simulation, and is not an extension of sequential simulation languages like Maisie or SIMSCRIPT III (RICE et al. 2005). Like other modern simulation languages (L'Ecuyer and Buist 2005), LAMPS can be displayed both graphically and as a rule-set.

LAMPS is based on hierarchical Petri Nets (Jensen 1992) and Colored Petri-Nets (cf. (Vojnar 1997)). The former introduced the idea to structure so-called places (states) and actions hierarchically, the latter introduced the idea of typed tokens. LAMPS extends petri-nets by the concept of agents.

LAMPSSys extends classical agent-based simulation (Moss and Davidsson 2001) by rigorously modelling all effects and events as actions of agents. Furthermore, LAMPSSys handles scheduling (i. e. when and in which order actions are executed) by grouping agents into so-called clans (see below).

LANGUAGE CONCEPTS AND FEATURES

Basic Constructs

LAMPS uses the following concepts from hierarchical and colored Petri nets:

- Places hold states, i.e. in LAMPSSys they contain subsets of the agent attributes. The contents of a place is called token and can be of an arbitrarily complex structured type. Places can contain several tokens of different or identical types.

- Actions describe the effects that agents apply to themselves or to other agents.

- Relations denote the links between places, actions, and agents. Relations correspond to the arcs in the Petri net model.

LAMPS introduces the additional concept of agents in the following way:

- Agents correspond to the conditions of Petri nets. An agent in LAMPS observes the set of places that have relations to the agent's actions. Based on these places the agent decides which actions are executed and which parameters should be used.

Agents extend guard functions (Esser 1997) and enabling functions (Schoef 1995). These serve as preconditions that are checked before an action is executed. Agents are explained in more detail below.
Figure 1 depicts a simple example using the four basic concepts.
LAMPS is inherently parallel due to its core of high-level petri nets. Agents live in parallel, and each agent can execute several parallel actions. All actions whose conditions are true in a given cycle are executed. This is also the main difference to flow-charts, because there are several tokens per place, and several places can be filled simultaneously.
All basic constructs can be recursively encapsulated. For example, places can be combined to so-called super-places. An action can be recursively defined as a LAMPS process (with places and other actions), as long as the interface (i. e. the incoming and outgoing places) of the action and the process are identical. This way, a process can be modelled and viewed on different detail levels.
Since LAMPS introduced agents, also agents can be recursively encapsulated. A group of agents can be aggregated into an agent (say, a group of soldiers into an infantry unit). The level of detail can be modulated even at runtime (see section below).

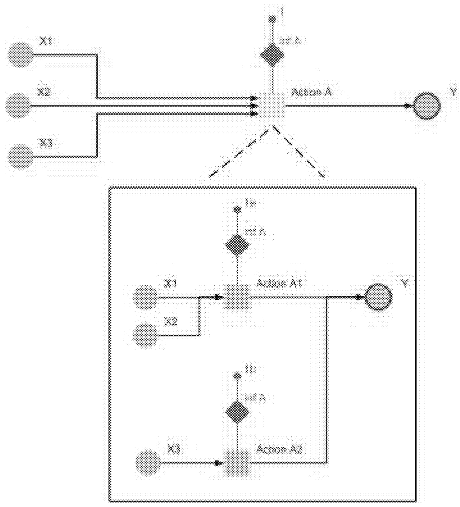


Figure 1: A LAMPS fragment using the four basic concepts. The agent *Inf A* observes the place *Enemy spotted*. If the place contains a token, the agent executes action *Combat*, which sends a token into the place *Resistance broken*.

Features

While LAMPS is not restricted to a time model, LAMPSSys is discrete-time and thus discrete-event. Time-modelling is not trivial in high-level petri-nets (cf. (Vojnar 1997)). Thus, in the LAMPSSys framework the execution is cycle-based with a global clock. In each cycle all agents can execute one action. Long actions are modelled as a series of consecutive actions of duration dt. Synchronisation is achieved via places and conditions.
In the approach, time scaling is possible. In each cycle, each agent proposes a duration for its action. The simulation engine selects the minimum of the proposed durations and sets dt accordingly. To illustrate this, assume that an agent is about to execute an action that needs a coarse granularity. This could be for example an action that moves a soldier from one position to a distant position, proposing $dt = 5min$. Assume that another agent is about to execute a more detailled action in the time-cycle, for example, shooting at an enemy, proposing $dt = 0.01sec$. In this case, the simulation will set the general dt to $0.01sec$, so that every action in this time cycle is executed in this granularity.

AGENT MODEL

Single agents

One of the main ideas of the LAMPS framework is to model every active entity as an agent. This can include humans, software agents, IT systems, and even environment objects like bridges. Each agent is autonomous and cannot be directly influenced by another agent. For example, if a soldier agent detonates a bomb at a bridge, the bridge agent is informed about the detonation. It then evaluates the effects on its state itself, eventually resulting in damage.

An agent consists of attributes and an interface. The former comprises the state of the agent, its behavior and its information about the environment. The interface is the protocol to interact with the environment.

The state of agents is represented as a tree of attributes. Thus, attributes can be handled hierarchically. Attributes are inherited if an agent's type is a subtype of another agent type.

Attributes can have the special type *behavior*. These behavior attributes contain LAMPS process descriptions that specify the behavior of an agent. These processes are ordered in terms of priority. Basic processes that describe physical events are executed before higher processes take place. For example, if an agent is in the state $falling$, then the physical event of falling is executed and overrides other more cognitive behaviors that could decide about movement.

Agents offer the following interface to their environment: The *tick* operator triggers the agent to execute its action for the cycle. The *read* operator then reads data from its communication channels (see below). After performing its decision, the agent can write data via the *write* operator to its communication channels.

Another important operator is the *rescale* operator which triggers the agent to aggregate itself with several agents to a super-agent, to disaggregate itself into several agents, or to rescale itself into another level of abstraction. Rescaling usually involves replacing the attribute set that describes the agent (and thus also its behavior).

Agent Clans

Agents can be grouped together into so-called clans. Every clan has one designated agent who serves as clan-chief. Clans are used to structure the relations between agents. Thus clans are a concept that facilitates modelling. Furthermore, as explained below, they increase performance.

Interaction-clans group agents together that can interact with each other in certain ways. For example, an interaction-clan can include those agents that can execute action type T and that can be effected by actions of type T. This way, checking which agents are affected by an action A can be limited to those that are in the same interaction-clan as the agent that executed A. Communication is a type of action. Interaction-clans can implement the common communication frameworks, such as blackboard, peer-to-peer, publish-and-subscribe.

Another important aspect for which clans are used is that of timing or scheduling. As already mentioned, the LAMPSSys framework is based on difference equations with adaptable time intervals dt. That is, a global time with adaptable time resolutions is used. In order to simulate different parts of the simulation at different time resolutions, agents can be grouped into scheduling clans. Members of the same scheduling clan use the same time resolution. This is achieved by the fact that the clan-chief triggers the *tick* operator of each agent in the scheduling clan. If the clan-chief triggers the agents only in certain conditions, also event-based scheduling can be realized.

A breeder-clan is responsible for instantiating agents. In a given breeder-clan, there exists a type-hierarchy of agents. An agent-type inherits its attributes (and thus also its behavior) from its father-type. The clan-chief can instantiate agents of each type. Furthermore, there exist so-called templates for instantiating agents. Templates not only specify the name and datatypes of attributes, but also fill them with default values.

LAMPSSYS ARCHITECTURE

LAMPSSys extends the Flip-Tick-Architecture (FTA) (Richter 1999), which has its roots in the JANUS project developed at the Gesellschaft fuer Mathematik und Datenverarbeitung (GMD) (Beyer and Smieja 1994). At its core, FTA is a design paradigm for scalable distributed systems that exhibit a priorily unknown dynamic characteristics as well as disturbances and inaccuracies which are difficult, if not impossible, to model in a closed-form mathematical approach.

LAMPSSys is based on the concept of clans, which are groups of agents. It comprises three classes of entities: agents, clans, and tags. A clan A is formed by a set of individual agents a_j.

$$A = \{a_1, \ldots, a_i, \ldots, a_k\}$$

Each actor a_j is composed of a set of typed attributes U_j, whose value assignment determines the agent's state, together with an action function f_j, which entails all operations that can be performed by the agent.

$$a_j = (U_j = (u_{1,j}, \ldots, u_{m,j}), f_j)$$

Structural information concerning agents, i.e. names and types of attributes, is described via agent types. Formally, we have $type(a_j) = t$, if and only if agent a_j is of type t. The system supports agent templates that can be used to store prototypical value assignments. Thus, an individual agent can be created either by instantiation of its type, or by copying from a pre-defined template.

The principle of autonomy of agents forbids the direct manipulation of internal data structures and behaviors of other agents. Consequently, all interactions between agents are handled via messages. Formally, a message N_v is comprised of a number of attributes:

$$N_v = \{u_{1,v}, \ldots, u_{l,v}\}$$

As agents, messages are typed entities. The set of all message types is given by

$$N = \{N_1, \ldots, N_h\}.$$

Upon receiving a request, the agent is able to analyze its content and to decide whether it wants to comply. The basic unit of execution is called a cycle. During one cycle, the agent reads its messages and triggers the appropriate actions, which might consist of writing messages to other agents.

A scheduling clan is a set of agents sharing a common pace, i.e. all elements of a clan have the same time resolution dt. This in turn implies that their cycles are synchronized and that the clan switches from cycle to cycle as regularly as the tick of a clock. It is important to note that different agents do not necessarily share the same time resolution. Instead, the architecture supports individual running speeds for agents. Moreover, time steps can vary from cycle to cycle. Thus, adaptive control of time increments can be realized (see below). This is particularly valuable for increasing the time resolution in the computation of dynamics equations for fast moving objects.

The messages used for inter-agent communication are called tags. Instead of setting up a direct communication with other agents, agents register with one ore more interaction-clan. They send their messages to a designated clan-chief of that clan. While clan-chiefs can in principle implement other communication protocols such as publish-and-subscribe or blackboard, LAMPSSys most often uses so-called tag-boards for discrete-time simulation. For event-based simulation other communication protocols are used. Tag-boards serve as the functional units for handling messages. In formal terms, a tag-board forms a medium M_n for message exchange, while a LAMPSSys system is capable of supporting multiple media:

$$M = \{M_1, \ldots, M_m\}$$

A tag-board consists of two sides. One is write-only and contains all tags sent to the board in time step t, whereas the other side is read-only and encompasses all tags written in time-step $t-1$. Analogously to agents, each tag-board has its own time resolution and thus its own cycle time. During a board cycle, the write-only side is flipped over. Thereby, the read-only part mirrors the tag content of the write part from the previous time-step. The write-only side is deleted after flipping. In this way, the lifetime of tags is effectively controlled by the time-scale of the pertaining board.

With this approach, fully synchronized (all agents and tag-boards share the same time resolution) as well as completely asynchronous systems (every agent and every board has its own time-scale) can be modeled in terms of the FTA.

The mathematical model of this agent architecture is a system of flexibly coupled inhomogeneous difference equations

$$D = \{D_1, \ldots, D_n\}$$

where each agent computes an equation from D. In this setting, the attribute list of an actor corresponds to the variable vector of the pertaining difference equation. Thus, equation D_i of agent a_i computes

$$U_i^{t+dt_i} = f_i(U_i^t)$$

where dt_i denotes the time step size of agent a_i. During the iteration of D, the clan of agents runs through a set of states $Z = \{Z^t\}$, where each of these system states is the union of attribute sets from all participating agents:

$$Z^t = (U_1^t, \ldots, U_i^t, \ldots, U_k^t)$$

Since the D_i are computed using potentially different dt_i, the involved agent attribute sets U_i^t might be undefined for certain t. In this case, either the U_i from the last time step of D_i is used, or U_i is explicitly recomputed. We thus have

$$Z = Z^0, \ldots, Z_{t^i}, Z^{t_j}, \ldots, Z^e$$

where

$$t_j - t_i = Min_{h=1}^{k} dt_k.$$

APPLICATIONS

The LAMPSSys framework is currently used in the following application fields.

ITSimBw is a system that simulates communication and IT aspects for modern warfare scenarios (Huegelmeyer et al. 2006). The idea of LAMPSSys to model every active element as an object allows the system to simulate the domain in a flexible and generic way. The language LAMPS is going to be the core of the data-mining approach in the future.

The aim of the IRRIIS project is to increase the dependability of large critical infrastuctures (Flentge and Steffens 2006). LAMPSSys is used to simulate the dependencies between telecommunication and electricity networks on different abstraction layers, including the physical, the management and the IT layer.

LAMPS was also used to model and implement the business logic of a large telecommunication company. The described processes are executed in a workflow-engine. In the future we aim to replace the workflow-engine by the LAMPSSys system in order to achieve maximal performance and parallelism.

CONCLUSIONS AND FUTURE WORK

We have introduced the LAMPSSys framework and have shown how processes are modelled using the LAMPS language. Furthermore we have outlined the agent model that allows a generic approach to modelling active elements in the domain. The behavior of agents is directly implemented as LAMPS process. Using the clan structure, it is possible to schedule agents and to structure agents into meaningful units. Future work includes using LAMPS for process traces that can be used for data-mining.

REFERENCES

Bagrodia R.L. and Liao W.T., 1994. *Maisie: A Language for the Design of Efficient Discrete-Event Simulations. IEEE Transactions on Software Engineering*, 20, no. 4, 225–238.

Beyer U. and Smieja F., 1994. *Janus: A society of agents*. Tech. rep., GMD, Sankt Augustin, Germany. GMD Report No. 840.

Esser R., 1997. *An Object Oriented Petri Net Language for Embedded System Design.* In *Proceedings of the 8th International Workshop on Software Technology and Engineering Practice (STEP)*. IEEE Computer Society Press, 216–223.

Flentge F. and Steffens T., 2006. *IRRIIS - A new European Project to Increase CII Dependability. European CIIP Newsletter*, 4, no. 1.

Healy K.J. and Kilgore R.A., 1997. *Silk : A Java-based Process Simulation Language.* In *Proceedings of the 1997 Winter Simulation Conference*. ACM, 475–482.

Huegelmeyer P.; Steffens T.; and Zoeller T., 2006. *Specifying and simulating modern warfare scenarios with ITSimBw.* In L.F. Perrone; F.P. Wieland; J. Liu; B.G.L.D.M. Nicol; and R.M. Fujimoto (Eds.), *Proceedings of the 2006 Winter Simulation Conference*. ACM.

Jensen K., 1992. *Coloured Petri Nets: Basic concepts, analysis methods and practical use, Monographs in Theoretical Computer Science*, vol. 1: Basic Concepts. Springer, Berlin.

L'Ecuyer P. and Buist E., 2005. *Simulation in Java with SSJ.* In *Proceedings of the 2005 Winter Simulation Conference*. ACM, 611–620.

Moss S. and Davidsson P., 2001. *Multi-Agent-Based Simulation.* Springer, Berlin.

RICE S.V.; Marjanski A.; Markowitz H.M.; and Bailey S.M., 2005. *The Simscript III Programming Language for Modular Object-Oriented Simulation.* In *Proceedings of the 2005 Winter Simulation Conference*. ACM, 621–630.

Richter G., 1999. *Flip-tick architecture: A cycle-oriented architecture for distributed problem solving.* Tech. rep., GMD, Sankt Augustin, Germany. GMD Report No. 19.

Schoef S., 1995. *A distributed simulation engine for hierarchical petri nets.* In *Simulation verteilter Systeme und paralleler Prozesse*. 153–159.

Teo Y.; Tay S.; and Kong K., 1998. *Structured Parallel Simulation Modeling and Programming.* In *Proceedings of the 31st Annual Simulation Symposium*. IEEE Computer Society Press, 135–142.

Vojnar T., 1997. *Various Kinds of Petri Nets in Simulation and Modelling.* In J. Stefan (Ed.), *Proceedings of 31st Spring International Conference on Modelling and Simulation of Systems MOSIS '97*. vol. 1, 227–232.

Weiss G., 2000. *Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence.* The MIT Press.

FORMAL INFRASTRUCTURE FOR VERIFICATION OF EPISTEMIC PROPERTIES OF MULTI-AGENT SYSTEMS

M. Bagić, M. Kunštić
Department of Telecommunications
Faculty of Electrical Engineering and Computing
University of Zagreb
Croatia
e-mail: marina.bagic@fer.hr, marijan.kunstic@fer.hr

KEYWORDS
ACTL, Model Checking, Multi-agent System, Verification of Epistemic Properties.

ABSTRACT

Verification of multi-agent systems (MAS) is a huge challenge, especially for those systems where security and safety are of major importance. Verification detects faults, defects and drawbacks in an early stage of software development. Here, we give a formal model for verification of MAS by means of model checking technique. We extend the existing Action Computation Tree Logic (ACTL) with epistemic operators in order to reason about knowledge properties of MAS. We introduce new operators for manipulation on agent's actions with data. We explain their syntax and semantics for our ACTL-er (ACTL for Epistemic Reasoning), and provide some algorithms to obtain epistemic properties from the model state space.

INTRODUCTION

This paper in a certain way is a continuation of the work presented in the papers [2, 3]. It basically reasons on concurrent systems verification by model checking via OBDD diagrams [6]. However, it does not deal with intelligent agents, so the main contribution of our paper is the extension of such an approach towards multi-agent problem domain. Also, the most important issue to be dealt here is the verification of multi-agent knowledge, or more precisely, dealing with epistemic properties of multi-agent systems. We extend the existing Action Computation Tree Logic (ACTL) towards Action Computation Tree Logic for Epistemic Reasoning (ACTL-er). We give its syntax and semantics with the emphasis on epistemic operators which manipulate on agent's epistemic properties. First we add data to actions from [3], and then divide these actions into tree colours; **weak**, **strong** and **forget** action.

Similar work is presented in papers like [9]. However, this approach to epistemic temporal verification are different than ours since they use CTL (Computation Tree Logic) as the basic logic, not ACTL. Temporal operators are $\mathbf{U}$, $\mathbf{X}$, $\mathbf{G}$, $\mathbf{F}$ while we modify these operators onto $\mathbf{U}_\kappa$, $\mathbf{X}_\kappa$, $\mathbf{G}_\kappa$, $\mathbf{F}_\kappa$, according to epistemic actions in our ACTL-er logic. Also, we add the $\mathbf{W}_\kappa$ as a new operator. Our basic definition of a knowledge is the same as in these papers. So, our verification of the knowledge contains similar operators over the knowledge propositions. Our approach is mainly different since it put the emphasis on the dynamic epistemic properties verification because it is based on actions reasoning while the CTLK (CTL for knowledge) approaches the states of the system. We take MTS (Mixed Transition System) [7] as a model for the system while CTLK takes Kripke structure as a model.

The other important related work is [4, 5, 7]. It is based on symbolical algorithms and OBDD's (Ordinary Binary Decision Diagrams) [6]. We chose this method for reduction the state space. While [4] use deontic interpreted systems for a basic model (similar to LTS), we use the parallel composition for modelling global states of the system as in [3].

ACTL EXTENSION FOR INTELLIGENT AGENTS

Multi-agent system (MAS) is well-represented with Labelled Transition System (LTS) [2, 3]. However, an LTS does not explicitly reason about knowledge. Therefore, in order to represent a MAS by an LTS, but to preserve MAS's knowledge qualities, we extend LTS towards the Kripke-alike structure, i.e. using both state and action labelling. Labelling transitions (actions and states) with data values, we enable knowledge modelling of MAS. Therefore, we define extended LTS structure, named Mixed Labelled Transition System (MTS), extending MTS from [7].

Definition 1: Mixed Labelled Transition System Mixed Labelled Transition System (MTS) is a 10-tuple:

$$\mathcal{M} = (\mathcal{S}, \mathcal{A}, \delta, \mathcal{K}, k_a, k_s, \mathcal{C}, c, \mathcal{P}, p)$$

where there are:

- S, a non-empty set of states
- $\mathcal{A}$, a finite, non-empty set of actions containing visible actions and silent action τ not visible to an external observer
- $\delta \subseteq \mathcal{S} \times \mathcal{A} \times \mathcal{S}$, the transition relation
- $\mathcal{K}$, a set of knowledge atomic propositions, or shortly, a knowledge atoms
- k_a, a an interpretation function on actions which assigns each action a set of knowledge atoms true for that action, $k_a : a \rightarrow 2^{\mathcal{K}}$, where $a \in A_\tau$
- k_s, an interpretation function on states which assigns each state a set of knowledge atoms true for that state, $k_s : \varphi \rightarrow 2^{\mathcal{K}}$, where $\varphi \in \mathcal{S}$
- $\mathcal{C}$, a set of actions colours (or types): $\{\chi, \kappa, \epsilon\}$
- c, a function that assigns each action a colour (or type) from the colour set $\mathcal{C}$,
  c: a $\rightarrow \mathcal{C}$
- $\mathcal{P}$, a set of agent's ports (or channels)
- p, a function that assigns each agent a set of ports (or channels) to communicate with another agents

We've introduced three colours (or types) of actions, denoted α, κ and ϵ. Compared to [2, 3] where each action formula $\{\chi\}\varphi$ "lives" only during one transition, we preserve knowledge obtained by the action $\kappa(\Delta)$ until action formula $\kappa(\neg\Delta)$ or $\epsilon(\Delta)$ occurs. Here, we denote the "usual" χ-action as $\alpha(\Delta)$-action enriched with data portion Δ, as **weak action**. And, $\kappa(\Delta)$-action as **strong action**. In order for an agent to forget the knowledge atoms Δ, we use $\epsilon(\Delta)$ action formula. This action formula deletes the set of agent's knowledge atoms Δ from the agent's memory, i.e. (s)he no longer knows whether these atoms are true or false. We refer to this action as **forget action**. This nomination is inspired by the actual meaning of the action formulae.

An action formula $\kappa(\neg\Delta)$ contains the same set of knowledge atoms as in $\kappa(\Delta)$ but with the opposite Boolean values. Thus, we strongly distinguish the $\{\alpha(\Delta)\}\varphi$ and $\{\kappa(\Delta)\}\varphi$ action formulae. We use $\{\alpha(\Delta)\}\varphi$ action formula in the same sense as defined in [2, 3], only with data values added. And, we use $\{\kappa(\Delta)\}\varphi$ action formula to add new knowledge to be preserved for a few consecutive action performances after this one.

An ACTL-er formula may include:

- constants: *true* and *false*
- action variables: $\alpha \in A_\tau$
- standard Boolean operators: $\neg, \wedge, \vee, \Leftrightarrow$
- *path quantifiers*:
  - **E**, "there exists a path"
  - **A**, "for all paths"
- *temporal operators*:
  - **U**, "until"
  - **W**, "unless" or "weak until"
  - **X**, "for the next transition"
  - **F**, "for some transitions in the future"
  - **G**, "for all transitions in the future".

Syntax and Semantics of ACTL-er

According to our definition of a MTS, we define new kind of ACTL-vp, a ACTL for Epistemic Reasoning, or shortly ACTL-er. ACTL-er syntax and semantics is defined over the MTS. The main difference with the above defined ACTL-vp is the introduction of action colours. Particularly, the epistemic coloured actions. The standardly defined ACTL [3], or ACTL-vp previously defined does not distinguish actions. An action formula is true while is occurring. There are no other side effects besides the transition to a new state. A data-action formula is a similar in the meaning that all the proposition atoms that are assigned to an action are true during the action performance. They are not true after the transition passes.

Definition 2: ACTL for Epistemic Reasoning Syntax Let χ, β, φ, and γ be an *data-action formula*, a *colour-action formula*, a *state formula*, and a *path formula*, respectively, iff they meet the following syntactic rules:

$$\chi ::= c\,!\,\beta \,|\, c?\;\beta \,|\; \tau(\beta) \,|\, \neg\chi \,|\, \chi \vee \chi'$$

$$\beta ::= \mathit{true} \,|\, \kappa(\Delta) \,|\, \epsilon(\Delta) \;|\, \alpha(\Delta) \;|\, \neg\beta \,|\, \beta \vee \beta'$$

$$\varphi ::= \mathit{true} \,|\, \neg\varphi \,|\, \varphi \wedge \varphi' \,|\; \mathbf{E}\gamma \,|\, \mathbf{A}\gamma$$

$$\gamma ::= \{\chi\}\varphi\, \mathbf{U}\, \{\chi'\}\varphi' \,|\; \{\chi\}\varphi\, \mathbf{W}\, \{\chi'\}\varphi' \,| \\ \{\chi\}\varphi\, \mathbf{U}_\kappa\, \{\chi'\}\varphi' \,|\; \{\chi\}\varphi\, \mathbf{W}_\kappa\, \{\chi'\}\varphi'$$

$h \models c\,!\,\beta$	iff c is outgoing $\wedge\, \beta = \mathit{true}$
$h \models c\,?\,\beta$	iff c is incoming $\wedge\, \beta = \mathit{true}$
$h \models \tau(\Delta)$	iff $h = \tau \wedge \Delta = \mathit{true}$
$h \models \neg\chi(\Delta)$	iff $h \not\models \chi(\Delta)$
$h \models \chi(\Delta) \vee \chi'(\Delta')$	iff $h \models \chi(\Delta) \vee h \models \chi'(\Delta')$

Table 1: Data-action Semantic Rules of ACTL for Epistemic Reasoning

Definition 3: ACTL for Epistemic Reasoning Semantics Let $\mathcal{M} = (\mathcal{S}, \mathcal{A}, \delta, \mathcal{K}, k_a, k_s, \mathcal{C}, c, \mathcal{P}, p)$. Satisfaction of data-action formula χ by an action $h \in \mathcal{A}$

(written $h \models \chi$), colour-action formula β by colour-action a, state formula φ by a state $p \in \mathcal{S}$ ($p \models \varphi$), a path formula γ by a finite fullpath π (written $\pi \models \gamma$), and a path formula γ by an infinite fullpath σ (written $\sigma \models \gamma$) in a MTS $\mathcal{M}$ is given inductively by the semantic rules given in tables 1, 2, 3 and 4.

$a \models true$	always
$a \models \alpha(\Delta)$	iff $a = \alpha \wedge \Delta = true$
$a \models \epsilon(\Delta)$	iff $a = \epsilon \wedge \Delta = true$
$a \models \kappa(\Delta)$	iff $a = \kappa \wedge \Delta = true$
$a \models \neg\alpha(\Delta)$	iff $a \not\models \alpha(\Delta)$
$a \models \alpha(\Delta) \vee \alpha'(\Delta')$	iff $a \models \alpha(\Delta) \vee a \models \alpha'(\Delta')$

Table 2: Colour-action Semantic Rules of ACTL for Epistemic Reasoning

Further, we provide with the algorithms for epistemic operators implementation by ACTL for each of the operator.

ALGORITHMS FOR EPISTEMIC OPERATORS IMPLEMENTATION

$\mathbf{U}_\kappa$ Epistemic Operator

To give a stronger insight into the given semantics, we also explain the formulae with the graphical notation where each relevant state and transition is labelled with the appropriate notation.
Let us assume when performing a $\kappa(\Delta)$-action an agent delivers a portion of knowledge into a new state. If the opposite knowledge atom was participated before that moment, i.e. the ¬atom was known to an agent, after $\kappa(\Delta)$-action an atom takes place, and vice versa.

$p \models true$	always
$p \models \neg\varphi$	$p \not\models \varphi$
$p \models \varphi \wedge \varphi'$	$p \models \varphi \wedge p \models \varphi'$
$p \models \mathbf{E}\gamma$	iff $\exists\pi : p = st(\pi, 0) \wedge \pi \models \gamma$
	or $\exists\sigma : p = st(\pi, 0) \wedge \sigma \models \gamma$
$p \models \mathbf{A}\gamma$	iff $\forall\pi$ and $\forall\sigma$
	$\pi : p = st(\pi, 0)$
	$\sigma : p = st(\pi, 0) \wedge \sigma \models \gamma$

Table 3: State Semantic Rules of ACTL for Epistemic Reasoning

Now we take a special insight into the performance of κ-action. This action has brought us to an introduction of new temporal operators, i.e. we use U_κ, F_κ, X_κ, G_κ and W_κ instead of U, F, X, G and W temporal operators for κ-actions verification. Here we ommit the send-receive policy, i.e. we do not explicitly reason on $c!$ or $c?$ part of the agents communication. We build κ-temporal operators in order to embrace the total agent(s) state space.

U_κ Epistemic Operator
Formula $\{\kappa(\Delta)\}\ \mathbf{U}_\kappa\ \{\kappa'(\Delta')\}$ is satisfied on a fullpath iff the first transition is $\kappa(\Delta)$-transition and neither $\kappa(\neg\Delta)$-transition, nor $\epsilon(\Delta)$-transition occurs until $\kappa'(\Delta')$-transition happens. Then, $\kappa'(\Delta')$ is valid through the next set of path's states until $\kappa'(\neg\Delta')$-transition, or $\epsilon(\Delta')$-transition occurs. The effect of $\mathbf{U}_\kappa$ operator expressed with ordinary $\mathbf{U}$ operator is as follows:

$$\{\kappa(\Delta)\}\mathbf{U}_\kappa\{\kappa'(\Delta')\} = (\{\alpha(\Delta)\}\mathbf{U}(\{\alpha'(\Delta') \wedge \epsilon(\Delta)\} \\ (\{\alpha'(\Delta')\mathbf{U}(\{\alpha'(\neg\Delta') \vee \epsilon(\Delta')\})$$

This formula makes the difference between the two operators obvious. They both express the meaning of usual ACTL $\mathbf{U}$ operator, except the $\mathbf{U}_\kappa$ adds some postoperational effects. We needed such an effects in order to preserve new knowledge from the transition following the first one.

X_κ Epistemic Operator
Formula $\mathbf{X}_\kappa\{\kappa(\Delta)\}$ is satisfied on a fullpath iff its first transition is $\{\kappa(\Delta)\}\varphi$-transition. The states following after this transition are also Δ-states until $\epsilon(\Delta)$ or $\kappa(\neg\Delta)$ occurs on the same path. The effect of $\mathbf{X}_\kappa\{\kappa(\Delta)\}$ operator is expressed with ordinary $\mathbf{X}$ and $\mathbf{U}$ operators as follows:

$$\mathbf{X}_\kappa\{\kappa(\Delta)\} = \mathbf{X}\{\alpha(\Delta)\}\ (\{\alpha(\Delta)\}\mathbf{U}\{\alpha(\neg\Delta \vee \epsilon(\Delta)\})$$

G_κ Epistemic Operator
Formula $\mathbf{G}_\kappa\{\kappa(\Delta)\}$ is satisfied on a fullpath iff its first transition is $\{\kappa(\Delta)\}\varphi$-transition and for all transitions after it nor $\kappa(\neg\Delta)$-action, neither $\epsilon(\Delta)$-action ever occurs. The effect of $\mathbf{G}_\kappa$ operator is expressed with ordinary $\mathbf{G}$ operator, where $\kappa(\Delta)$-action on the right hand side has the same meaning as the $\chi(\Delta)$, as follows:

$$\mathbf{G}_\kappa\{\kappa(\Delta)\} = \mathbf{G}\{\alpha(\Delta)\}$$

F_κ Epistemic Operator
Formula $\mathbf{F}_\kappa\{\kappa(\Delta)\}$ is satisfied on a fullpath iff there exists a $\kappa(\Delta)$-action on it. The states following after this transition are also Δ-states until $\epsilon(\Delta)$ or $\kappa(\neg\Delta)$ occurs on the same path. The effect of $\mathbf{F}_\kappa$ operator is expressed with ordinary $\mathbf{F}$ and $\mathbf{U}$ operators as follows:

$$\mathbf{F}_\kappa\{\kappa(\Delta)\} = \mathbf{F}\{\alpha(\Delta)\}(\{\alpha(\Delta)\} \\ \mathbf{U}\{\alpha(\neg\Delta) \vee \epsilon(\Delta)\}$$

W_κ Epistemic Operator
Formula $\kappa(\Delta)\ W_\kappa\ \kappa'(\Delta')$ is satisfied on a fullpath iff $\kappa(\Delta)\ U_\kappa\ \kappa'(\Delta')$ is satisfied on it or $G_\kappa\{\kappa(\Delta)\}$ is satisfied on it.

ALGORITHMIC RESOLVING EPISTEMIC OPERATORS

In this section we reason on semantics of the epistemic ACTL operators. MTS abstract diagrams are given in order to visualise the effect of each operator on MTS set of states. This approach was strongly inspired by [3]. According to the above definitions appropriate δ-transition sets are defined: $\delta_{\kappa(\Delta)} \subseteq \delta$ denotes the set of all $\kappa(\Delta)$-transitions from δ, $\delta_{\neg\kappa(\neg\Delta)}$ denotes the set of all transitions from δ which are not $\kappa(\neg\Delta)$-transitions, $\delta_{\kappa(\neg\Delta)} \subseteq \delta$ denotes the set of all $\kappa(\neg\Delta)$-transitions from δ, $\delta_{\neg\kappa(\Delta)} \subseteq \delta$ denotes the set of all transitions different than $\kappa(\Delta)$ from δ, $\mathcal{S}_\Delta \subseteq \mathcal{S}$ denotes the set of all Δ-states from $\mathcal{S}$ and $\mathcal{S}_{\neg\Delta} \subseteq \mathcal{S}$ denotes the set of all states from $\mathcal{S}$ which are not Δ-states.

EU$_\kappa$ Operator

Let $\mathcal{X}_{\kappa(\Delta),\kappa'(\Delta')}$ be the set of states where ACTL formula $\mathbf{E}\kappa(\Delta)\mathbf{U}_\kappa\kappa'(\Delta')$ holds. We calculate it by the following fixed point formula:

$$\begin{gathered}\mathcal{X}_{\kappa(\Delta),\kappa'(\Delta')} = \mathbf{lfp}\mathcal{Z}.(\{(q \in \mathcal{S}_\Delta |\ \exists a \in \mathcal{A}_\tau \exists q' \in \mathcal{S}_{\Delta'} \\ (q,a,q') \in \delta_{\kappa'(\Delta)'} \\ \vee \exists a \in \mathcal{A}_\tau \exists q' \in \mathcal{Z}\, (q,a,q') \in \delta_{\neg(\kappa(\neg\Delta)\vee\epsilon(\Delta))} \\ \vee \exists a \in \mathcal{A}_\tau \exists q' \in \mathcal{Z}\ \ (q,a,q') \in \delta_{\kappa(\Delta)}) \\ \vee(q' \in \mathcal{S}_{\Delta'} | \exists a \in \mathcal{A}_\tau \exists q' \in \mathcal{Z}\ \ (q,a,q') \in \\ \delta_{\neg(\kappa(\neg\Delta')\vee\epsilon(\Delta'))})\})\end{gathered}$$

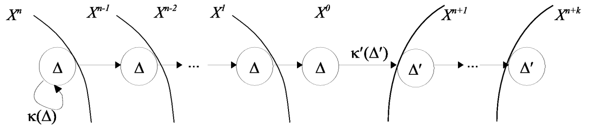


Figure 1: Resolving **EU** Operator

The least fixed point can be effectively determined by calculating the sequence of set of states $X^0_{\kappa(\Delta),\kappa'(\Delta')}$, $X^1_{\kappa(\Delta),\kappa'(\Delta')}$, ... where $X^i_{\kappa(\Delta),\kappa'(\Delta')} \subseteq \mathcal{S}$ is given as follows:

$$\begin{gathered}X^0_{\kappa(\Delta),\kappa'(\Delta')} = \{q \in \mathcal{S}_\Delta |\ \exists a \in \mathcal{A}_\tau \exists q' \in \mathcal{S}_{\Delta'} : \\ (q,a,q') \in \delta_{\kappa'(\Delta)'}\}\end{gathered}$$

$$\begin{gathered}0 < i < n : X^i_{\kappa(\Delta),\kappa'(\Delta')}\ X^{i-1}_{\kappa(\Delta),\kappa'(\Delta')} \bigcup \\ \{q \in \mathcal{S}_\Delta | \exists a \in \mathcal{A}_\tau \exists q' \in X^{i-1}_{\kappa(\Delta),\kappa'(\Delta')} : (q,a,q') \in \\ \delta_{\neg(\kappa(\neg\Delta)\vee\epsilon(\Delta))}\}\end{gathered}$$

$$\begin{gathered}X^n_{\kappa(\Delta),\kappa'(\Delta')} = X^{n-1}_{\kappa(\Delta),\kappa'(\Delta')} \bigcup \\ \{q \in \mathcal{S}_\Delta |\ \exists a \in \mathcal{A}_\tau \exists q' \in X^{n-1}_{\kappa(\Delta),\kappa'(\Delta')} : (q,a,q') \in \delta_{\kappa(\Delta)}\}\end{gathered}$$

$$\begin{gathered}1 < i < k : X^{n+i}_{\kappa(\Delta),\kappa'(\Delta')}\ X^{n+i-1}_{\kappa(\Delta),\kappa'(\Delta')} \bigcup \\ \{q' \in \mathcal{S}_{\Delta'} | \exists a \in \mathcal{A}_\tau \exists q' \in X^{n+i-1}_{\kappa(\Delta),\kappa'(\Delta')} : (q,a,q') \in \\ \delta_{\neg(\kappa(\neg\Delta')\vee\epsilon(\Delta'))}\}\end{gathered}$$

Suppose that ACTL formula $\mathbf{E}\kappa(\Delta)\mathbf{U}\kappa'(\Delta')$ holds in the initial state s_0 and let $X^n_{\kappa(\Delta),\kappa'(\Delta')}$ be the first set in the sequence $X^0_{\kappa(\Delta),\kappa'(\Delta')}$, $X^1_{\kappa(\Delta),\kappa'(\Delta')}$, ... which contains state s_0. Then, a witness exists which is a finite path beginning in Δ-state s_0 and consisting of a sequence of n $\neg(\kappa(\neg\Delta) \vee \epsilon(\Delta)$-actions followed by $\kappa'(\Delta')$-action. To construct this witness, we start in a state s_0 with $\kappa(\Delta)$-action and follow a path where $\forall i \in [1, n-1] :\ s_i \in X^{n-i}_{\kappa(\Delta),\kappa'(\Delta')} \backslash X^{n-i-1}_{\kappa(\Delta),\kappa'(\Delta')}$, $\forall i \in [1, n-1] : (s_i, a_{i+1}, s_{i+1}) \in \delta_{\neg(\kappa(\neg\Delta)\vee\epsilon(\Delta))}$, and $(s_n, a_{n+1}, s_{n+1}) \in \delta_{\kappa'(\Delta')}$, $\exists k \in \mathcal{N} : (s_{n+k}, a_{n+k+1}, s_{n+k+1}) \in \delta_{\neg(\kappa(\neg\Delta')\vee\epsilon(\Delta'))}$.

EG$_\kappa$ Operator

Let $\mathcal{X}_{\kappa(\Delta)}$ be the set of states calculated by the following fixed point formula:

$$\begin{gathered}\mathcal{X}_{\kappa(\Delta)} = \mathbf{gfp}\mathcal{Z}.(\{q \in \mathcal{S}_\Delta \vee \mathcal{S}_{dead} |\ \exists a \in \mathcal{A}_\tau \exists q' \in \mathcal{Z} : \\ (q,a,q') \in \delta_{\kappa(\Delta)} \\ \vee \exists a \in \mathcal{A}_\tau \exists q' \in \mathcal{Z} : (q,a,q') \in \delta_{\neg(\kappa(\neg\Delta)\vee\epsilon(\Delta))}\})\end{gathered}$$

Then, $\mathcal{X}_{\kappa(\Delta)}$ is the set of all states where ACTL formula $\mathbf{EG}_\kappa\kappa(\Delta)$ holds.

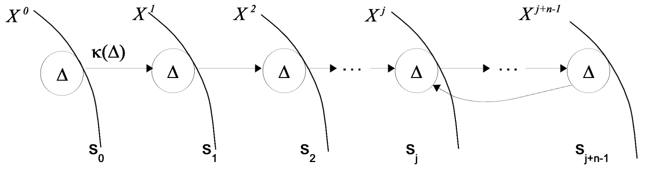


Figure 2: Resolving **EG** Operator

Suppose that ACTL formula $\mathbf{EG}_\kappa\kappa(\Delta)$ holds in the initial state s_0. A witness is trivial if s_0 is a deadlocked. Otherwise, a witness exists which is a finite path beginning in Δ-state s_0 and $\kappa(\Delta)$-action followed by a set of actions $\neg(\kappa(\neg\Delta) + \epsilon(\Delta))$ leading to a deadlocked state, or an infinite path beginning in Δ-state s_0 and $\kappa(\Delta)$-action followed by a set of actions $\neg(\kappa(\neg\Delta) + \epsilon(\Delta))$. To construct this witness we start in state s_0 and follow a path where (s_0) and initial $\kappa(\Delta)$-action are followed by a set of actions $\forall \in [1, j-1] : (s_i, a_{i+1}, s_{i+1}) \in \delta_{\neg(\kappa(\neg\Delta)+\epsilon(\Delta))}$ until a state appears which is on this cycle or the deadlocked state is reached.
Checking whether a state $s \in \mathcal{X}_{\kappa(\Delta)}$ is on a cycle can be done by calculating a sequence of sets of states. So, let us symbolically find the satisfying states, taking one step at a time.

$$X_0 = \{q \in S_\Delta \,|\, a \in A_\tau\ q' \in S_\Delta\ \ : (q,a,q') \in \delta_{\kappa(\Delta)}\}$$

$$\begin{gathered}X_n = X_{n-1} \bigcup\ \{q' \in S_\Delta \vee S_{dead} \,|\, a \in A_\tau\ q \in X_{n-1} \\ : (q,a,q') \in \delta_{\neg(\kappa(\neg\Delta)+\epsilon(\Delta))}\}\end{gathered}$$

CONCLUSION

In this paper we have given the formal infrastructure for verification of MAS based on model checking with symbolic approach (OBDD's). We have given a syntax and semantics of ACTL for Epistemic Reasoning in order to reason about agent's knowledge. With ACTL-er logic we are able to write formulae in the form of "does this property holds in the model". The implemented model checker should provide us with an answer. Therefore, for the future work, model checker should be tested. Also, different approaches to parallel composition of agents can be explored in order to obtain optimal inter-agent communication and cooperation.

REFERENCES

[1] R. Fagin, J. Y. Halpern, Y. Moses, M. Y. Vardi, Reasoning About Knowledge, The MIT Press, Cambridge Massachusetts, London England, 2003

[2] R Meolic, T. Kapus, Z. Brezocnik, Verification of concurrent systems using ACTL, In Applied informatics: proceedings of the IASTED international conference AI'2000, M. H. Hamza, ed., Anaheim, Calgary, Zrich, IASTED/ACTA Press, Innsbruck, Austria, pages 663-669, February 14-17, 2000

[3] R. Meolic, T. Kapus, Z. Brezocnik, An Action Computation Tree Logic With Unless Operator, Proceedings of the 1st South-East European workshop on formal methods SEEFM 2003, Thessaloniki, Greece, pp 100-114, November 20, 2003

[4] F. Raimondi, A. Lomuscio, Automatic verification of deontic and epistemic properties of multi-agent systems by model checking via OBDD's, Proceedings of ECAI 2004, Valencia, August 2004

[5] F. Raimondi, A. Lomuscio, Symbolic model checking of multiagent systems via OBDD's: an algorithm and its implementation, Proceedings of AAMAS04, New York, August 2004

[6] R. E. Bryant. Graph-based algorithms for boolean function manipulation. IEEE Transactions on Computers, C-35(8), 1986

[7] C. Pecheur and F. Raimondi, Symbolic model checking of logics with Actions, to appear in Proceedings of the Fourth Workshop on model checking artificial intelligence (MoChArt 2006), Springer Verlag LNAI

[8] P. Kefalas, M. Holcombe, G. Eleftherakis, M. Gheorge, A Formal Method for the Development of Agent Based Systems, In Intelligent Agent Software Engineering, V.Plekhanova (eds), Idea Group Publishing Co., pp.68-98, 2003

[9] M. Kacprzak, *From bounded to unbounded model checking for temporal epistemic logic*, Fundamenta Informaticae 62, pp. 120, 2003 IOS Press

$\pi \models \varphi\{\chi(\Delta)\}\,\mathbf{U}\,\{\chi'(\Delta')\}\varphi'$
iff $st(\pi,0) \models \varphi \wedge \exists i \in [1,|\pi|] : (act(\pi,i) \models \chi'(\Delta')$
$\wedge st(\pi,i) \models \varphi') \wedge \forall j \in [1,i-1] : (act(\pi,j) \models \chi(\Delta)$
$\wedge st(\pi,j) \models \varphi)$

$\pi \models \varphi\{\kappa(\Delta)\}\,\mathbf{U}_\kappa\,\{\kappa'(\Delta')\}\varphi'$
iff $st(\pi,0) \models \Delta \wedge act(\pi,1) \models \kappa(\Delta)\exists i \in [1,|\pi|]$
$: (act(\pi,i) \models \kappa'(\Delta') \wedge st(\pi,i) \models \Delta') \wedge \forall j \in [1,i-1] :$
$(act(\pi,j) \not\models \kappa(\neg\Delta) \vee \epsilon(\Delta) \wedge st(\pi,j) \models \Delta) \wedge \forall k > 1$
iff $act(\pi,k) \not\models \kappa(\neg\Delta) \vee \epsilon(\Delta))st(\pi,k) \models \Delta')$

$\sigma \models \varphi\{\chi(\Delta)\}\,\mathbf{U}\,\{\chi'(\Delta')\}\varphi'$
iff $st(\sigma,0) \models \varphi \wedge \exists i \in [1,|\sigma|] : (act(\sigma,i) \models \chi'(\Delta')$
$\wedge st(\sigma,i) \models \varphi') \wedge \forall j \in [1,i-1] : (act(\sigma,j) \models \chi(\Delta)$
$\wedge st(\sigma,j) \models \varphi)$

$\sigma \models \varphi\{\kappa(\Delta)\}\,\mathbf{U}_\kappa\,\{\kappa'(\Delta')\}\varphi'$
iff $st(\sigma,0) \models \Delta \wedge act(\sigma,1) \models \kappa(\Delta)\exists i$
$: (act(\sigma,i) \models \kappa'(\Delta') \wedge st(\sigma,i) \models \Delta') \wedge \forall j \in [1,i-1] :$
$(act(\sigma,j) \not\models \kappa(\neg\Delta) \vee \epsilon(\Delta) \wedge st(\sigma,j) \models \Delta) \wedge \forall k > 1$
iff $act(\sigma,k) \not\models \kappa(\neg\Delta) \vee \epsilon(\Delta))st(\sigma,k) \models \Delta')$

$\pi \models \{\chi\}\varphi\,\mathbf{W}\,\{\chi'\}\varphi'$
if $\pi \models \{\chi\}\varphi\,\mathbf{U}\,\{\chi'\}\varphi'$ or if
$\forall i \in [1, len(\pi)] st(\pi,i) \models \varphi \wedge\ act(\pi,i) \models \chi$

$\pi \models \{\chi\}\varphi\,\mathbf{W}_\kappa\,\{\chi'\}\varphi'$
if $\pi \models \{\chi\}\varphi\,\mathbf{U}_\kappa\,\{\chi'\}\varphi' \vee$
$st(\pi,1) \models \varphi \wedge act(\pi,1) \models \kappa(\Delta) \wedge$
$\forall j > i\,(act(\pi,j) \not\models \kappa(\neg\Delta) \vee \epsilon(\Delta) \wedge\ st(\pi,j) \models \Delta)$

$\sigma \models \{\chi\}\varphi\,\mathbf{W}\,\{\chi'\}\varphi'$
if $\sigma \models \{\chi\}\varphi\,\mathbf{U}\,\{\chi'\}\varphi'$ or if
$\forall i \geq 1 : st(\sigma,i) \models \varphi \wedge\ act(\sigma,i) \models \chi$

$\sigma \models \{\chi\}\varphi\,\mathbf{W}_\kappa\,\{\chi'\}\varphi'$
if $\sigma \models \{\chi\}\varphi\,\mathbf{U}_\kappa\,\{\chi'\}\varphi' \vee$
$st(\sigma,1) \models \varphi \wedge act(\sigma,1) \models \kappa(\Delta) \wedge$
$\forall j > i\,(act(\sigma,j) \not\models \kappa(\neg\Delta) \vee \epsilon(\Delta) \wedge\ st(\sigma,j) \models \Delta)$

Table 4: Path Semantic Rules of ACTL for Epistemic Reasoning

PETRI NETS FORMALISM

PETRI NET - BASED PROJECT SCHEDULING METHODS: ADVANTAGES AND SHORTCOMINGS

Konstantinos Kirytopoulos[1] Viktor Diamantas[2] Vrassidas Leopoulos[3] Christos Dimadis[4]

University of the Aegean
Department of Financial and Management Engineering
Fostini 31
Chios, Greece

National Technical University of Athens
Department of Mechanical Engineering
Iroon Polytechniou 9
Athens, Greece

E-mail: [1]kkir@central.ntua.gr, [2]vdiamant@mail.ntua.gr, [3]vleo@central.ntua.gr, [4]mc01023@central.ntua.gr

KEYWORDS
Project Management, Scheduling, PERT, Petri Nets.

ABSTRACT

Scheduling of construction projects has been a major issue since construction industry exists. It defines project activities, their duration, the construction sequence, and the resources requirements trying to come up with the most efficient project plan. There are generally accepted methods for modelling project duration such as the Critical Path Method, the Program Evaluation and Review Technique, the Graphical Evaluation and Review Technique, and the Monte Carlo Simulation (MCS). Several authors have proposed the use of Petri Net (PN) based scheduling methods. This paper describes such a method as far as the estimation of project duration is concerned and compares the deriving results with those of PERT. The study is enhanced by a case study, which reveals that under certain circumstances the differences between the two tools might be of minor importance, thus the literature research findings stating that simulation models give more reliable results than PERT are downgraded.

1. INTRODUCTION

Project management is of great importance for the Construction industry. Construction projects are complex, depending mostly on their size and have many inherent uncertainties, that derive from the unique characteristics of each project, the diversity of resources and activities, as well as external factors. The project management community has defined all these uncertainty factors as risks (PMI 2004). According to PMI (2004) a project risk is *an uncertain event or condition that, if it occurs, has a positive or negative effect on at least one project objective*. Moreover, risks are divided in discrete risks and the embedded duration uncertainty (Diamantas et al. 2006) and they can affect the criteria against which the success of a project is usually measured, namely time (schedule), cost and quality (or performance). By the term ‘embedded uncertainty’ the authors describe the volatility in the duration of an activity which exists due to many complex factors that most of the times the project management team does not want or cannot analyse further than assigning a statistical distribution to the duration. The paper focuses on the embedded duration uncertainty in order to provide a comparative study, and come up with certain conclusions concerning the use of Petri Nets based scheduling methods in construction projects versus typical PERT. Petri Nets were chosen as an alternative since they have a series of benefits for project scheduling. Among others, Petri Nets can handle efficiently the use and scheduling of resources, thus they are regarded as a valid project scheduling tool (Pagnoni 2006). Resource scheduling is a long-standing problem that “burdens” project managers until now. Another argument for the selection of such a tool is that Petri Nets may perform “clock independent project management” by providing to the project management team a clear and sound representation of tasks interrelationships and resource needs (Pagnoni 2006).

PERT is a well known method with proven value in managing complex projects. It was developed in 1958 by the US Navy Special Projects Office as part of the Polaris mobile submarine launched ballistic missile project (Malcolm et al. 1959).

Petri Nets are graphical and mathematical modelling tools that can be used to perform static and dynamic modelling of existing or new systems (Tatsiopoulos and Leopoulos 1999). Carl Petri is generally considered as the originator of the concept of Petri Nets (Petri 1966). Since then the use and study of Petri Nets has increased considerably (Leopoulos 2000).

Stochastic Petri Nets (SPNs) were introduced in 1980 (Symons 1980) as a formalism for the description of Discrete Event Dynamic Systems (DEDS). With the aim of extending the modelling power of SPNs, Generalised Stochastic Petri Net (GSPNs) were defined by Marsan et al. (1984). That definition was later improved to better exploit the structural properties of the modelling paradigm (Marsan et al. 1987). GSPNs include two classes of transitions: exponentially distributed timed transitions, and immediate transitions. Since transitions in a GSPN can have deterministic and stochastic delays, incorporation of risk and uncertainty in the estimates of tasks duration is possible (Sawhney et al. 2003).

Figures 1 and **2** illustrate a simple project schedule. **Figure 1** utilises the CPM method to formulate the project network and **Figure 2** shows the equivalent PN-based project network.

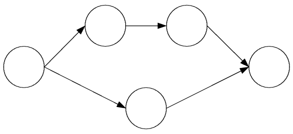


Figure 1: CPM based Project Network

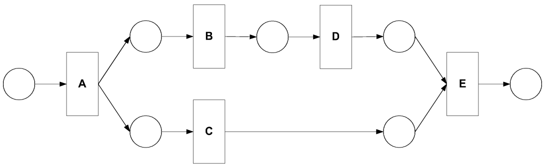


Figure 2: PN-based project Project Network

2. PROBLEM FORMULATION AND METHODOLOGY

The study aimed to identify advantages and shortcomings of a PN-based project scheduling method and followed a four step methodology. The first step was the literature review of scheduling focusing on PERT and PN-based project scheduling methods in the construction industry. Construction industry was selected due to its great interest on project scheduling techniques. However, the study can be fairly fitted to any kind of industry elaborating project scheduling. The theoretical findings are summarised in **Section 3**.

The second step was the development of the "project model". In order to develop the model, the major activities of the construction project selected and the construction sequence were identified. Initially, a Gantt chart of the project was created through a project scheduling software. Then, the duration of each activity's duration was estimated. The three point estimate approach was used, which required the estimation of an optimistic (minimum), a most likely, and a pessimistic (maximum) value for the duration of each activity. These values were estimated by the project team, based on past experience acquired from similar projects, and the use of the corporate memory. Once the model for PERT analysis had been completed the model for the PN-based project scheduling method was developed. Transitions and places, were used to model the activities of the construction project. Hierarchical transitions allowed for a systematic top-down breakdown of the construction project, that followed the codification of the project's Work Breakdown Structure (WBS). The PN model was developed using Artifex software tool, developed by Faber Software.

The third step was the conduction of PERT analysis.

The PN-based project scheduling method took place afterwards. The three point estimation of activity duration performed during step 2 was used. The simulation ran 1000 iterations and the results were recorded in a spreadsheet. Additionally, the best fitting distribution to the results was defined.

The final step of this method was the comparison of the results generated by PERT against the results generated by the PN-based project scheduling method, in order to identify whether the results validate the theoretical findings or not.

3. THEORETICAL FINDINGS

The authors conducted a literature review for project scheduling focusing on PERT and PN-based project scheduling methods. Scientific papers, conference proceedings, case studies and books dealing with scheduling have been explored and conclusions regarding the use of PERT and PNs have been drawn. The following paragraphs summarise, the theoretical conclusions.

The main conclusion is that while PERT is a simple and straight forward method, requiring less time and resources than PN-based project scheduling methods, it suffers from major limitations. Diamantas et al. (2006) state that PERT, limited by the constraints of the Central Limit Theorem, is obliged to model the duration of every activity of a project with the same statistical distribution, while the statistical distributions of all the tasks are assumed to be independent. Additionally, it ignores all "sub-critical" paths, taking into account a unique critical path as it is determined by deterministic calculations. Finally, PERT has difficulties in dealing with risks beyond the inherent uncertainties of activity duration.

As far as PN-based scheduling methods are concerned, it is concluded that they do not face PERT's limitations. Those methods are not based on the Central Limit Theorem, thus activity durations may be modelled by different distributions. Likewise, no statistical independence is assumed for activities' durations. Furthermore, PN-based project scheduling methods take into account all possible paths within the project.

PNs have modelling versatility and are effective in modelling a stochastic and dynamic system (Sawhney 1997) such as a construction project and allow for the handling of projects for which uncertainty is very important (van der Aalst 1996), while the representation of the precedence relationships, as well as the modelling of precedence constraints can be done easily (Jeetendra et al. 2005). They have a hierarchical structure (top-down breakdown) that allows the development of a very abstract top-level with increasing model detail in the lower levels (Jeetendra et al. 2005). Furthermore the scheduling process can be simplified by using PNs modular features that allow the reuse of repetitive modules (Sawhney et al. 2003).

Besides, by using PNs the discrete risks and the embedded duration uncertainty that a project is susceptible to can be incorporated and thus many external factors that affect the project can be taken into account (Sawhney et al. 2003). Moreover PNs allow the concise modelling of many real situations that can occur in a project, such as deadlock states and conflicts (Tuncel and Bayhan 2006). Finally simulation with PNs can take into account the dynamic allocation of recourses (Sawhney et al. 2003) and all the constraints that can be related with resources such as partial allocation, substitution of recourses or mutual exclusivity.

Despite these major advantages PN-based project scheduling methods suffer from major drawbacks that hinder their use by project managers. They are far more complex than PERT, requiring more time, special software, computer time and trained individuals.

4. CASE STUDY

4.1 Description of the project

The presented project was a medium scale construction project located in the Greek island of Chios that was undertaken by a major international contractor based in Greece. It dealt with the construction of several independent residential buildings.

The Gantt chart of the project (**Figure 3**) consists of 68 activities. The pessimistic (maximum), most likely and optimistic (minimum) duration of each activity has been estimated by the project team, taking into account its knowledge and past experience acquired from similar projects. Each activity's duration was modelled in the Petri Net as a triangular distribution, since this was assumed to be the most reliable distribution for a construction project (Rentizelas et al. 2006). The deterministic duration of the project (CPM method), based on the critical path and the estimated most likely duration for each activity, was calculated to be 291 days. The typical approach of PERT (PERT-typical) uses the original estimators for the Beta distribution.

But, the most interesting, since it allows for comparison with the PN-based project model (where activities' durations have been inserted as triangular distributions), was the PERT-triangular. In the PERT-triangular approach the authors have reached the final PERT's normal distribution by working with mean values and standard deviations of the triangular and not the Beta distributions, as these figures have resulted from Palisade's @Risk Software Tool.

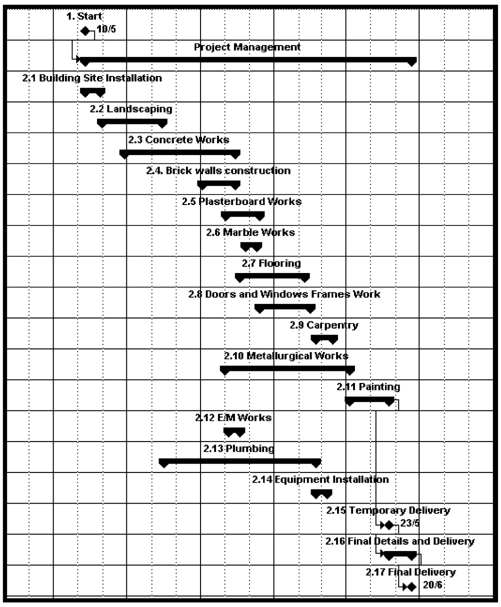


Figure 3 Project's Gantt Chart

For reasons of brevity the complete PN-based model is not presented in this paper. Instead of it, the top level (**Figure 4**) and two second level modules (project summary tasks) are presented.

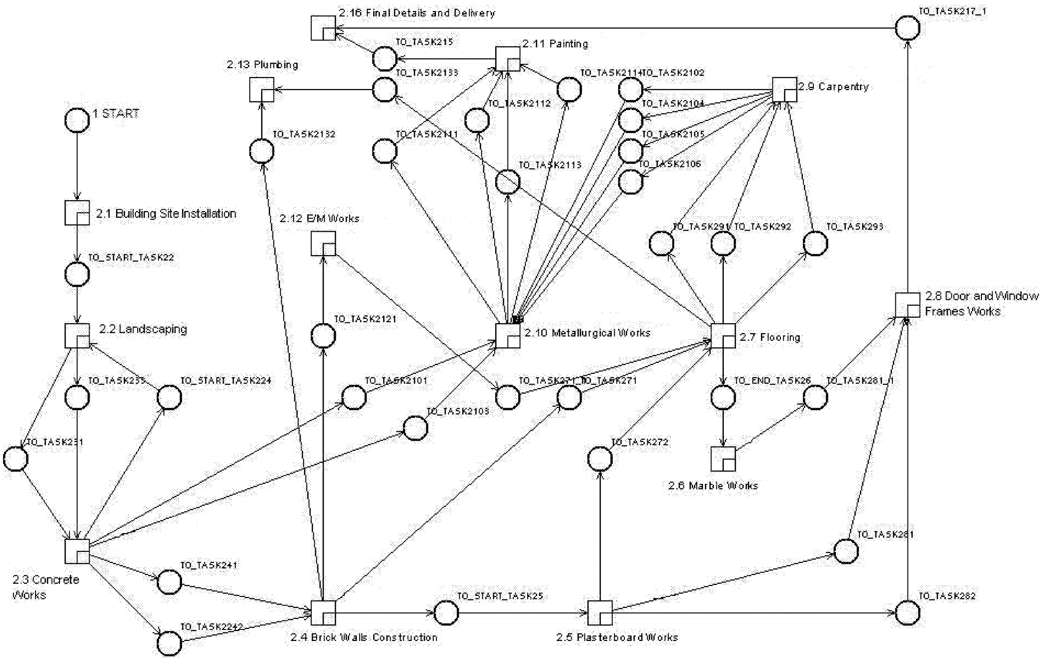


Figure 4 PN-based Model Top Level

Figures 5 and **6** show two second level modules, each representing a single summary task, landscaping and flooring, respectively. These modules have higher detail and it's easier to observe the tokens' movement through the tasks. The activity IDs shown in these figures follow the Work Breakdown Structure (WBS) codification of the deterministic model of the project.

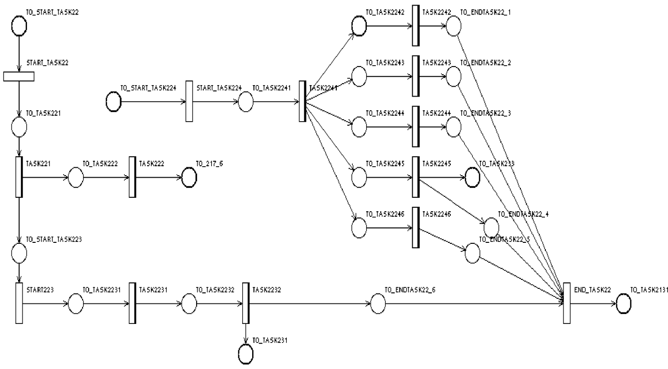


Figure 5 PN-based Model Landscaping

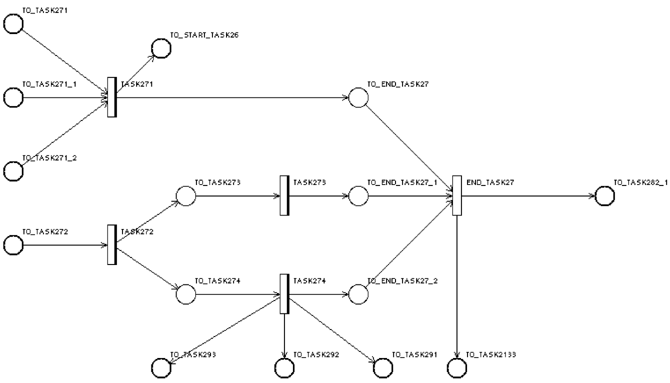


Figure 6 PN-based Model Flooring

4.2 Analysis of the results

The results of the deterministic (CPM) approach, 291 days, corresponds to a 4.4% confidence level in PERT-triangular and is slightly lower than the minimum value (291.3 days) provided by the PN-based project scheduling method. **Table 1** presents the, three; best fitting distributions to the results of the PN-based project scheduling method, ranked by the P-value.

Table 1 Best Fitting Distribution of PN Results

Distribution	P-value
Weibull (α=2.92, β=19.77 Shift = +291.31)	0.9653
BetaGeneral (α_1=7.44; α_2=8.52; min=283; max=761; 337.87)	0.8085
Normal(308.95; 6.54)	0.4631

The best fitting distribution "Weibull (α=2.92, β=19.77 Shift = +291.31)" was used in the analysis that followed. In **Figure 7** the cumulative density function of this distribution is compared with the simulation results; the gray scale area represents the simulation results while the black line represents the best fitting distribution's cumulative density function.

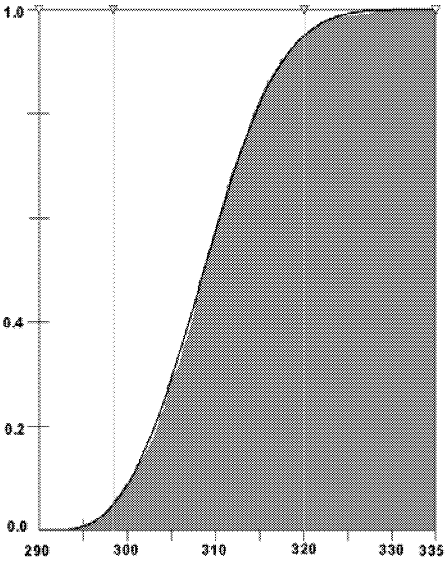


Figure 7 Comparing best fit distribution with simulation results

In **Figure 8** the cumulative density function that derives from PERT-triangular is compared with the cumulative density function that derives from the PN-based project scheduling method, as well as the PERT-typical.

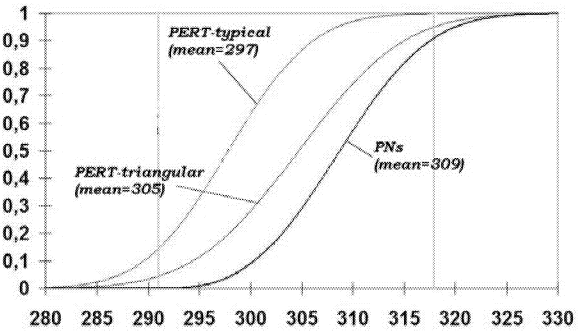


Figure 8 PN, PERT-triangular, PERT-typical distributions

As shown in **Figure 8** the PN approach (μ= 308.94, σ= 6.57) provides more conservative results than PERT-triangular (μ=304.7, σ=8.05), for a confidence level up to 99%. However, the differences in the mean values; 2.7% between PERT-typical and PERT-triangular, 4.0% between PERT-typical and PN approach, and 1.4% between PERT-triangular and PN approach, are remarkably low.

5. CONCLUSIONS – LIMITATIONS – FURTHER RESEARCH

The objective of this study was to compare the results of a PN-based project scheduling method with those of PERT. PNs were chosen, among other tools that enable simulation, as they may be expanded while research continues to cover related issues of resource levelling – allocation.

The results of the literature review indicated that a major PERT limitation is that it does not take into account sub-critical paths in the estimation of the project's duration.

This problem is overcome from Petri Nets, as well as from any simulation tool.

Furthermore, simulation has a proven value that is the revealing of critical tasks. Through simulation, activities can be sorted by the times that each one has participated in the critical path. While the number of iterations of the simulation gets bigger the project manager will be more confident about the activities that have to be closely monitored.

However, very often, project teams are short of time, thus in need for a quicker scheduling tool. Moreover, due to lack of data, project management teams tend to use the same distribution for activities' duration modelling (thus loosing a PERT limitation), which most of the times is the triangular or beta distribution. In the case study presented here, the outcome of the PERT versus that of the Petri Nets model is practically insignificant (1.4% difference between the mean value of PERT and Petri Nets model, respectively – 4 days difference in a project of 304 days mean duration) to the project manager. Although the confidence levels for certain project durations are quite different for the two approaches (refer to **Figure 8**), results indicate that for the project under investigation it might not worth the time needed to invest in a Petri Nets model approach, if resource levelling/allocation is not in the agenda. In other words, even if it is well established in the literature that a simulation tool would provide more reliable information than PERT, it seems that the differences produced in the case study are of minor importance for the project manager – decision maker. So under very specific circumstances simulation might give less benefits than usual.

In addition PNs require a relatively extended amount of time for their development and there is not a project scheduling specific PN software tool to assist the analyst (i.e. transform a Project Plan to a Petri Net). Such a tool would render Petri Nets competitive against other modelling tools used in project scheduling.

Finally, as long as resource levelling/allocation issues are not addressed, the use of a less complex simulation tool instead of PNs would provide a more simple approach requiring less time and resources. Otherwise, PNs would have been the most effective tool, since they achieve an efficient modelling of project resources' usage.

No safe conclusions can be drawn from a single case (although same indications exist for other thirty projects that the authors have analysed). However, the importance of this finding is the indication that under certain circumstances (not only with the absence of sub-critical paths) PERT technique might be enough for schedule analysis, thus valuable time and analysis costs can be saved from the project.

The authors have already targeted their research towards developing a method for resource scheduling using PN-based methods.

REFERENCES

Diamantas, V., Kirytopoulos, K. and Leopoulos V., 2006, "Simulation in Project Scheduling: Why PERT is not enough", *Proceedings of the 3rd Future Business Technology Conference*, The European Simulation Society, Athens, Greece, pp. 64-70.

Jeetendra V.A., Krishnaiah Chetty O.V, and Prashanth J., 2000, 'Petri Nets for Project Management and Resource Levelling', International Journal Of Advanced Manufacturing Technology, Vol. 16, No. 7, pp. 516–520.

Koller G., 1999, "Risk assessment and decision making in business and industry: a practical guide", CRC Press.

Leopoulos V., 2000, "Modelling and simulation of an extended enterprise using Generalised Stochastic Petri Nets", *In proceedings of ICT in Logistics and Production Management*.

Malcolm DG, Rooseboom JH, Clark CE, and Fazer W., 1959, "Application of a technique for research and development program evaluation", *Operations Research*, Vol. 7, pp. 646-669.

Marsan M.A., Balbo G., and Conte G., 1984, "A class of generalized stochastic Petri nets for the performance analysis of multiprocessor systems", *ACM Transactions on Computer Systems, 2(1)*.

Marsan M.A., Balbo G., Chiola G., and Conte G., 1987, "Generalized stochastic Petri nets revisited: Random switches and priorities", *In Proceedings of International. Workshop on Petri Nets and Performance Models*, pp. 44–53, Madison, WI, USA, IEEE-CS Press.

Pagnoni, A. 1990, *Project engineering: Computer-oriented planning and operational decision making*, Springer Verlang.

Petri C., 1966, "Communication with Automata", *Technical Report*, Rome Air Dev. Centre, New York, RADC-TR-65-337.

PMI, 2004, "A Guide to the Project Management Body of Knowledge", Project Management Institute, USA.

Rentizelas, A., Tziralis, G. and Kirytopoulos, K. 2006, 'An integrative approach to investment appraisal: Simulating the risk of an optimum decision', *Proceedings of the 3rd Future Business Technology Conference*, The European Simulation Society, Athens, Greece, pp. 27-34.

Sawhney A., Mund A., and Chaitavatputtiporn T., 2003, "Petri Net-Based Scheduling of Construction Projects", *Civil Engineering and Environmental Systems*, Vol. 20, No. 4, pp. 255–271.

Sawhney A., 1997, "Petri Net - based Simulation of Construction Schedules", *Proceedings of the 1997 Winter Simulation Conference*, pp. 1111-1118.

Symons F.J.W., 1980, "Introduction to numerical Petri Nets, a general graphical model for concurrent processing systems", *Australian Telecommunications Research*, Vol. 14, No. 1, pp. 28-33.

Tatsiopoulos I. and Leopoulos V., 1999, "Hierarchical integration of enterprise modelling methodologies using Petri Nets", in Dimitris K. Despotis, Constantin Zopounidis (Eds), Integrating Technology and Human Decisions: Global Bridges into the 21st Century, 5th International Decision Sciences Institute Conference.

Tuncel G., and Mirac Bayhan G., 2006, "Applications of Petri nets in production scheduling: a review", *The International Journal of Advanced Manufacturing Technology*, pp.1-12, DOI 10.1007/s00170-006-0640-1.

van der Aalst W.M.P., 1996, "Petri Net - based scheduling", *OR Spectrum*, Vol. 18, Issue 4, pp. 219-229.

PAINTED PETRI NET AND FUNCTIONAL ABSTRACTION TO VISUALIZE DYNAMIC MODELING IN BIOLOGICAL MODELS

Simon Hardy and Pierre N. Robillard
Département de génie informatique
École Polytechnique de Montréal, P.O. Box 6079, Succ. Centre-Ville,
Montréal, H3C 3A7, CANADA
E-mail: {simon.hardy, pierre-n.robillard}@polymtl.ca

ABSTRACT

The new Painted Petri net and Functional Abstraction techniques are combined in a visualization approach aimed at representing the dynamic behavior of simulated complex Hybrid Functional Petri net models. It is a visual approach merging the Petri net model structure with simulation data. We developed this approach in order to enhance the comprehension of biological models behavior after simulation and the representation of results. The results are two dynamic views of a model. The first view focuses on the variation of the individual elements of the model. The second view aims at representing the variation of the model functional activities. The approach is illustrated with a model from systems biology: visualization of the dynamic behavior of a biochemical model obtained from the hybrid functional Petri net model of the CaMKII regulation pathway.

INTRODUCTION

The Petri net is a modeling approach that is used in biology to complete theoretical studies of different biological systems (Hardy and Robillard 2004). Qualitative studies exploit the mathematical formalism of Petri nets to determine the properties of a biological system and to validate a biological model (Heiner et al. 2004) while simulated Petri net models are the basis for quantitative studies. Different Petri net extensions, such as the colored, stochastic and hybrid extensions, have been used for biological modeling in various studies. A variant of this modeling approach, the Hybrid Functional Petri net (HFPN), was designed specifically for biological molecular modeling and has been successfully used to study a number of different biological systems (Matsuno et al. 2003, 2006). The software that has been developed for the modeling and simulation of the HFPN is the Genomic Object Net (GON) (Nagasaki et al. 2003). A HFPN model and the its simulation data are analyzed to understand the dynamics and the behavior of the modeled biological system. With the software GON, the simulation data can be visualized in real-time on selected plots during the simulation or it can be exported to files compatible with any spreadsheet program.

To obtain more realistic molecular models of biological systems, the models tend to become more complex. They contain more elements and more intricate interactions. This complexity raises new issues for modelers. Once the model is approximately completed, it has to be validated and an thorough analysis must be done so that the modeler grasps the detailed behavior of its model. This is one of the most important goals of the systems biology research field: to achieve a system-level understanding about biological systems. The simulation of complex models generate a great quantity of data and the analysis of this data can be an arduous and cumbersome task and a system-level understanding can be unachievable with traditional data representations.

Among the methodologies available based on modeling and simulation, some exploit information visualization and visual data mining to acquire system knowledge. Keim (2002) clearly stated the benefits of visual data exploration, which integrates human perceptual abilities in the interpretation of the data and is highly useful in the exploratory steps of data analysis. Some visualization methods exist for biological kinetic models based on ordinary differential equations. Rost and Kummer (2004) developed a visualization software tool, SimWiz, which produces a dynamic representation of the simulation data of biochemical networks by presenting the data in a comprehensive way and by preserving the topology of the modeled system. Other methods for visualizing Petri net dynamic behavior already exist. Three-dimensional visualization of the Petri net has been developed by Kindler and Páles (2004), but, in their work, a Petri net model represents real world objects: a train and a railway controller. Software is also available with GON, called Cell Animator, to create animations based on simulation data. For example, Cell Animator can be used to represent a cell with several enzymes and metabolites interacting with one another, their number and position varying according to the numerical values obtained from simulation. These animated representations are useful for rapidly communicating information to others who are unfamiliar with the model, particularly in a teaching context. In the type of problem that interests us, however, such representations are unsatisfactory or even impossible. A 3D-visualization of a model of cellular signaling pathways, where the Petri net entities are associated with animated objects, is not feasible because we are dealing with concentrations of diffused substances. An animation of moving molecules becomes quite confusing when tens, or even hundreds, of different substances are involved. More importantly, these animations do not incorporate the topology of the modeled signaling pathways, and, as a result, crucial information about the model needed to reach a system-level understanding is lost.

Since the human mind has an incredible capacity to detect structures and patterns in images (Müller and Schumann

2003), we have devised two visual dynamic representations of HFPN simulation data which integrate the structure and the dynamics of a system model and present the model at different levels of abstraction. The objective of our methods is to enhance the comprehension of a model behavior and the representation of the simulation data in order to provide insights leading to a system-level understanding of a dynamic model. We call the first representation the *painted Petri net*. It is a simple but effective representational approach merging the HFPN model structure with the simulation data. Readers already familiar with the colored Petri net must not confuse that particular extension of Petri net theory with the dynamic representation of systemic behavior presented in this paper. In the colored Petri net, coloration is an abstract concept to distinguish different data types inside a model. In the painted Petri net, tints or colors are used to visually indicate the temporal variation of the modeled entities represented by places. In this paper, we also explain how a second method called *functional abstraction* can generate representations at different abstraction levels with a modular and functional approach. The main idea behind functional abstraction is to group HFPN elements into subnets based on molecular function and to devise equations computing the subnets activity state from the simulation data. To view the complete system behavior, the activity variations of every subnets of a model are visualized in a single colored dynamic representation. In this paper, both approaches are illustrated with a model from systems biology: a hybrid functional Petri net model of the CaMKII regulation pathway. Finally, we discuss the possible applications of our dynamic representation and future work.

A VISUAL DYNAMIC REPRESENTATION OF THE HYBRID FUNCTIONAL PETRI NET MODEL BEHAVIOR WITH THE PAINTED PETRI NET

According to the Müller and Schumann (2003) taxonomy of methods of time-dependent data visualization, the painted Petri net is a dynamic representation of multivariate data with a continuous linear time axis. Being time-dependent means that the visual representation changes dynamically over time and is a function of time. Two aspects of the model representation have to be considered: how to represent the model *structure* or *topology* and how to represent the model *dynamics*. These two aspects correspond to the two data sources needed as input for the software implementing this method: a file containing the organizational specifications of the model (HFPN elements and interactions) and a file containing the simulation data.

Representation of the model structure

In HFPN modeling of molecular biology systems, places correspond to molecular substances and are represented as circles; transitions correspond to discrete events or to continuous chemical reactions and are represented as rectangles; and arrows link places to transitions and represent the relation between substances and chemical reactions. The time-varying data of interest are the numerical values associated with places. Those values mainly represent substance concentrations or numbers of molecules. The HFPN formalism includes discrete and continuous places, discrete and continuous transitions, and three arc types: normal, inhibitory and test. We refer the reader to Matsuno et al. (2003) for a complete introduction to the HFPN.

Keeping the Petri net structure for a dynamic representation is efficient for the modeler, as this individual is already accustomed to the model's abstracted elements and no transformation or rework is needed. One way to specify a Petri net model is to use a software with a graphical editor, like GON. The model is specified and its structure is created by arranging Petri net elements in a drawing space. The model structure can be saved in a XML format.

Representation of the model dynamics

A hybrid Petri net simulator like GON deals with discrete and continuous variables. To visualize the simulation data, plots and graphs are produced for any variable of the HFPN model. In molecular biology models, these are concentration graphs. When the simulation results of complex models are presented, only significant graphs illustrating important features of a system are displayed. However, if all the simulation data have to be explored and monitored, the modeler has to deal with one graph for each substance. This type of representation with graphs can rapidly become overwhelming for complex models involving many substances. For example, the case study model of this paper contains approximately one hundred substances. There are twice as many substances in a more complete model of this biological system, adding other signaling pathways to the CaMKII regulation pathway to form a network (Bhalla and Iyengar 1999), and there are five times as many substances in the latest enlarged version of the network model (Ma'ayan et al. 2005).

In our representation, we decided to use color to improve the visual aspect of this information visualization method. For our dynamic visual representation, the painted Petri net, we painted the simulation data on the HFPN model structure using an 11-color palette (visible light spectrum: from violet to red). Numerical values are normalized with equation (1).

$$T_{i,t} = \left\lceil 10\left(\frac{x_{i,t} - x_{i,\min}}{x_{i,\max} - x_{i,\min}}\right)\right\rceil \qquad (1)$$

In equation (1), $T_{i,t}$ is the tint of the substance i at time t, $x_{i,t}$ is the concentration or number of molecules of the substance i at time t, and $x_{i,min}$ and $x_{i,max}$ are the minimum and maximum values of the concentration or number of molecules of substance i. The variable i corresponds to a HFPN place and $x_{i,t}$ corresponds to the marking of a discrete place or the mark of a continuous place. The application of the round-up function and the multiplication by ten return an integer value from 0 to 10, which corresponds to one of the 11 discrete tints of the palette (0 is mapped to violet and 10 is mapped to red). By applying this formula to the simulation data of a model, we obtain a tint for each substance at every simulation step. It is then possible to juxtapose the tints, which reflect the dynamic behavior of the model, to the model structure. This is done by painting the places of the

Petri net model, as shown in Fig. 1. The painted structure of the model at a given simulation step is one frame of the dynamic system representation. The HFPN model painted with the normalized concentrations for every time step results in successive frames that are assembled into an animation. This creates a movie-like representation of the Petri net dynamics, which can only be produce when the simulation is completed so that minimum and maximum values are known. Fig. 1 shows a painted HFPN representation of the model of the Ca^{2+} calmodulin-dependent protein kinase II (CaMKII) regulation pathway of Bhalla and Iyengar (1999). The HFPN net model was edited and simulated with GON. The CaMKII regulation pathway model is part of a larger network in the hippocampal CA1 neuron, which has already been thoroughly studied for its role in neuronal synaptic plasticity. The model's details and composition, as well as its simulation results, are outside the scope of this paper. The reader should consult the referenced paper for more information about this biological system.

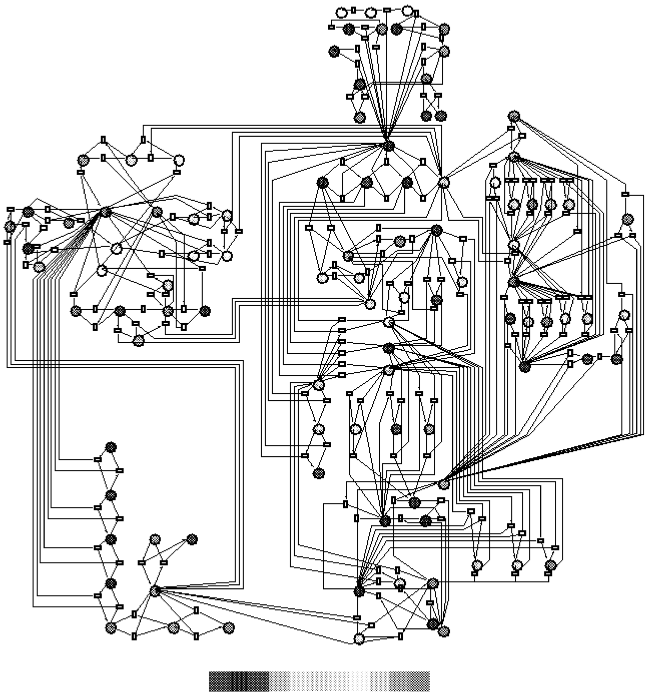


Figure 1: A frame of the dynamic representation of a painted HFPN molecular biology model, simulation step 68 s

HIGHLIGTHING THE HYBRID FUNCTIONAL PETRI NET MODEL BEHAVIOR WITH FUNCTIONAL ABSTRACTION

The behavior of biological systems is the result of the interactions between small units. The behavior of these small units is easily understood and can usually be predicted. However, the systemic behavior resulting from their complex interactions is far from being obvious without proper analyses. The painted Petri net creates a single viewpoint of the variations of every places of the model. The modeler can view the speed and synchronicity of certain changes in the model. The modeler can locate areas with an interesting or unexpected pattern of activity. Still, the link between the variations of the individual elements and the function of the small units of the system is not straightforward. A cellular signaling pathway, because of its information processing capabilities, shares similarities with an electronic circuit. In this analogy, the painted Petri net enables a view of the variation of the electric current in a circuit where each place is a connection. The painted Petri net is not a view of the variation of the activity of the different electronic components of the circuit. The functional abstraction enables this view of a system. The different steps to produce a view based on functional abstraction are:

1. Group HFPN places into subnets having a common function in the model. These subnets are called functional units.
2. Write equations involving the markings and marks of the subnet places from the simulation data. These activity equations describe the mechanism of functional units. They return a discrete value between 0 and 10 that is associated to a color. The value 0 stands for low activity (violet) and 10 for high activity (red).
3. Connect every functional units with arrows to create a new representation of the model. In molecular biology, we recognize two types of relationship between functional units: activation a inhibition. The new representation has two types of arc corresponding to these relationships.

We applied this methodology to the CaMKII regulation pathway model. Many functional units have been identified and their activity equations have been written. They are based on biological knowledge that is outside the scope of this paper. A color for every functional unit was computed at every time step. The result is another dynamic representation, or animation, of the model. Fig. 2 depicts four frames of a dynamic representation of the CaMKII regulation pathway model and its functional units. As we can see in Fig. 2, the Petri net model places and transitions are transformed into rounded rectangles linked by arrows. This view of the model has fewer elements and it is a coarse-grained abstraction of the network topology. The grouping approach is similar to the ideas of modular cell biology of Hartwell et al. (1999), who argued for the recognition of functional modules as a critical level of biological organization in the cell.

The malleability of the functional abstraction method enables a variety of subnet creation rules, which can create dynamic representations of a system at different levels of abstraction. In the model presented in this article, one organizational level is the module. Modules are pathway building blocks. They are represented as dotted rectangles in Fig. 2. Fig. 3 shows a representation of the model at the module level of abstraction. At this level, there is only one functional unit for each module. New equations describing the modules mechanism were devised and the emphasis is on the interactions between modules. For the CaMKII regulation pathway model, three levels of abstraction are enough to decompose the model into organizational levels. Any higher abstraction level would be useless for a model of this complexity, but for more complex models, additional abstraction levels could be useful.

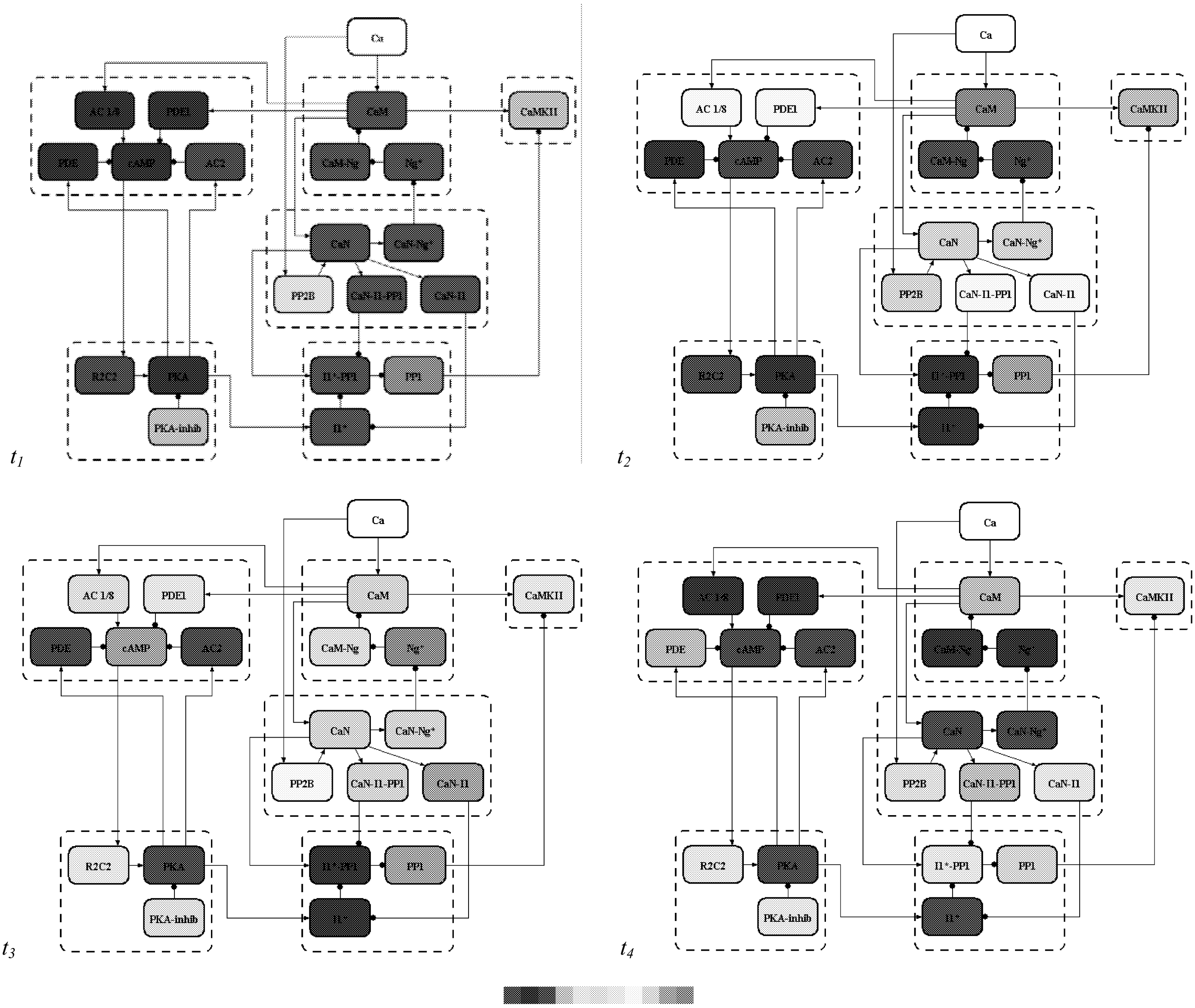


Figure 2: Four frames of the dynamic representation of the functional units derived from a Petri net biological model, simulation steps t_1 = 0 s, t_2 = 62 s, t_3 = 68 s and t_4 = 500 s

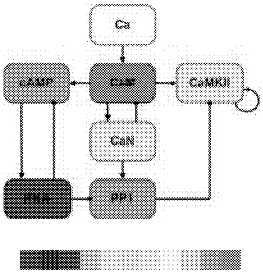


Figure 3: A frame of the dynamic representation of the higher-level modules derived from a Petri net biological model, simulation step 68 s

DISCUSSION AND CONCLUSION

The Painted Petri net and the Functional Abstraction techniques are the constituent parts of a new approach to visualizing simulation data and the dynamic behavior of modeled systems. An example of this approach was given with the visualization of the molecular biological model of the CaMKII regulation pathway. In systems biology, modeling and simulation are used to study system behavior, and so this visualization approach can be useful in this area of research. However, we are extending the use of this visualization method to any application domain where complex Petri net models are used to study model dynamics by simulation. The painted Petri net can be useful to modelers during the design of the model, as it can help to rapidly provide them with a picture of the model's dynamic behavior and help them to find the sources of problem or error. Usually, Petri net simulators represent the dynamics of the model with animations of discrete tokens being generated and consumed by the firing of discrete transitions or by displaying the real values of the markings of continuous places as continuous transitions are fired. Obtaining a global appreciation of the dynamics of a large Petri net model with these simulators is burdensome and inefficient, and the painted Petri net is a solution to this problem. We applied the painted Petri net to HFPN, but it can be adapted to any Petri net extensions that is used for simulation.

For the functional abstraction technique, we favor grouping rules based on function. Our main concern is to enable modelers to use their specific knowledge related to the modeling application domain. The cognitive effort of building a model gives to modelers the skill to identify the subnets and to mathematically characterize their mechanism. Each application domain is likely to have some specific abstraction rules defining the model organization and the appropriate abstraction levels. An automatic algorithm to identify functional units can be developed for a human engineered application domain. For example, in a model of an electronic system, every components have a known design and function, and their quantity is finite. In biology, the modeled systems are of a different nature. Evolution led to a great number of different designs. Moreover, the automatisation of the functional abstraction technique would transform it into an analysis method identifying the functional units of a model. The primary objective of the methods presented in this paper is to be a method to create new model representations precisely tailored to the modelers' investigations and concerns, not an analysis tool.

There are two advantages to the new views of simulation data created by functional abstraction. First, by reducing the number of elements in the dynamic representation, the graphical display is simplified. This advantage will become more valuable as models increase in complexity. Second, grouping different substances linked to the same biochemical functionality adds a systemic knowledge filter that can facilitate the visualization of system behavior. The viewers can navigate between the different organizational layers of the model, thus refining their system-level understanding of the model. An important concept in systems biology is emergent property, which is defined as a system-level characteristic resulting from the complex and nonlinear interactions of system elements and that cannot be predicted from what is known about these elements. In other words, we might easily understand the dynamics of the system elements taken separately, but when these elements are connected to form a complex network, it is impossible to predict the dynamics of the overall network. Emergent properties are nonintuitive, and systemic analysis through simulation is the main approach to understanding them. Indeed, the study of emergent properties was the main motivation behind the creation of the Painted Petri net and the Functional Abstraction techniques. By isolating specific functional activities and displaying their interactions in a dynamic representation, our visualization method can help modelers to detect and characterize the emergent properties of a model. Whether a network exhibits multistability, specific oscillatory patterns or synchronization, these complex properties may now be rapidly detected visually, revealing particular systemic properties of interest, which would then be worth exploring with other analytical methods.

The dynamic represention presented in this paper have been generated with a software prototype developed in java. At the moment, we do not intend to develop a complete software tool. This software would be rather simple and we prefer to pursue the development of the methodology first. In the near future, we will apply functional abstraction to a more complex biological model and we will investigate some abstraction levels where the automatic identification of cellular regulatory motifs, such as feedforward and feedback loops, could be implemented.

ACKNOWLEDGMENTS

This work was supported in part by NSERC grant A-0141 and an NSERC Postgraduate Scholarship.

REFERENCES

Bhalla, U.S. and Iyengar, R. 1999. "Emergent properties of networks of biological signaling pathways." *Science* 283, 381-387.

Hardy, S. and Robillard, P.N. 2004. "Modeling and simulation of molecular biology systems using petri nets: modeling goals of various approaches." *Journal of Bioinformatics and Computational Biology* 2(4), 595-613.

Hartwell, L.H., Hopfield, J.J., Leibler, S. and Murray, A.W. 1999. "From molecular to modular cell biology." *Nature* 402(6761), 47-52.

Heiner, M.; Koch, I. and Will, J. (2004), "Model validation of biological pathways using Petri nets, demonstrated for apoptosis", *Biosystems* 75(1-3), 15-28.

Keim, D.A. 2002. "Information visualization and visual data mining." *IEEE Transactions on Visualization and Computer Graphics* 8(1), 1 - 8.

Kindler, E. and Páles, C. 2004. "3D-Visualization of Petri net models: Concept and realization." In *Proceedings of the Applications and Theory of Petri Nets 2004* (Bologna, ITALY June 21-26). 3099, 464-473.

Ma'ayan, A., Jenkins, S.L., Neves, S., Hasseldine, A., Grace, E., Dubin-Thaler, B., Eungdamrong, N.J., Weng, G., Ram, P.T., Rice, J.J., Kershenbaum, A., Stolovitzky, G.A., Blitzer, R.D. and Iyengar, R. 2005. "Formation of regulatory patterns during signal propagation in a mammalian cellular network." *Science* 309(5737), 1078 - 1083.

Matsuno, H., Murakami, R., Yamane, R., Yamasaki, N., Fujita, S., Haruka, Y. and Miyano, S. 2003. "Boundary Formation by Notch Signaling in Drosophila Multicellular Systems: Experimental Observations and Gene Network Modeling by Genomic Object Net." In *Proceedings of the Pacific Symposium on Biocomputing* (Hawaii, USA, Jan. 3-7). 152-163.

Matsuno, H., Inouye, S.T., Okitsu, Y., Fujii, Y. and Miyano, S. 2006. "A new regulatory interaction suggested by simulations for circadian genetic control mechanism in mammals." *Journal of Bioinformatics and Computational Biology* 4(1), 139-153.

Müller, W. and Schumann, H. 2003. "Visualization methods for time-dependent data – an overview." In *Proceedings of the 2003 Winter Simulation Conference* (New Orleans, LA, USA Dec. 7-10). Vol. 1, 737-745.

Nagasaki, M., Doi, A., Matsuno, H. and Miyano, S. 2003. "Genomic Object Net: I. A platform for modeling and simulating biopathways." *Applied Bioinformatics* 2(3), 181-184.

Rost, U. and Kummer, U. 2004. "Visualisation of biochemical network simulations with SimWiz." *IEE Systems Biology* 1(1), 184 - 189.

A Meta-modeling Approach for Sequence Diagrams to Petri Nets Transformation within the requirements validation process

Adel Ouardani
Philippe Esteban
Mario Paludetto
Jean-Claude Pascal
Laboratoire d'Analyse et d'Architecture des Systèmes LAAS-CNRS
Université Paul Sabatier
7 Av. du Colonel Roche,
31077 Toulouse Cedex 4, France.
{ouardani, esteban, paludetto, jcp } @laas.fr

KEYWORDS
Meta-modeling, Petri nets, Sequence diagrams, Model transformation, Validation and Verification

ABSTRACT

This paper deals with the transformation of UML Sequence Diagrams into Petri Nets. The involved SD are those joined to the Use Cases dynamic description. In other words, they concern the requirements modeling. The approach is seen in a more general context of heterogeneous system design. It endeavors to integrate, early in the requirement modeling process, a formal model within a semi-formal one, in this case Petri nets with UML SD. The SD to PNs transformation is meta-modeling oriented within the MDA approach, specifically for PIM design. So, a meta-model for SD and one for PNs are first defined. Then, the meta-model of the transformation is done, and the transformation rules may be deduced. As a result, the approach allows a partial automation of the transformation, and subsequently the verification and validation of the system requirements.

1. INTRODUCTION

The Model Driven Architecture "MDA" proposed by the Object Management Group (OMG) is increasingly used to define an approach to software development based on modeling, and automated mapping of models to implementations. It becomes even so for the design of heterogeneous systems too, where several domains and trades are strongly involved. Its recent industrial success, along with its use in an ever greater number of industrial projects has prompted engineers and researchers to define and use MDA based approaches. The basic MDA concept involves defining a Platform Independent Model "PIM" and its automated mapping to one or more Platform-Specific models "PSM" from the Platform Dependent Model too (Frankel 2003). We defined a platform in accordance with such a concept for the virtual prototyping of the system to design. The virtual prototyping designed with an MDA view allows us to make a system V&V (Validation and Verification) by means of both, the formal verification and the system simulation. The aim is to tackle the design and implementation steps with requirements models free of faults (Kleppe et al. 2003).

The work presented in this paper can be located at the highest level of the PIM development when the customer requirements are translated into models. At the beginning, the PIM production is mainly based on the UML language. The today design trend entails to integrate or extend formal models early in the modeling process because of many benefits to do so, consistency check of the requirements models, system V&V or just to match with some PDM. In our approach the formal model is PNs because our PDM is also PNs based, and also we have a good expertise in using PNs in several ways (Esteban et al. 2005), the formal V&V, the operating of its structural properties as an aid for system structuring and the system simulation that results of the PNs transformation into VHDL-AMS (Albert et al. 2005) (see Fig. 1).

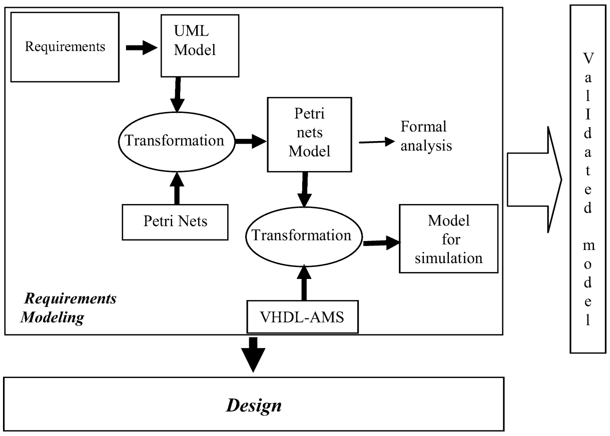


Fig.1. Requirements validation process

While the current OMG standards such as the Meta Object Facility "MOF" and the UML provide a well-established foundation for defining PIMs and PSMs, no such well established foundation exists for transforming UML PIMs into formal model for V&V purpose (Gerber et al. 2002), (Czarnecki and Helsen 2003) nor into PSMs. In our work the PIMs are related to the requirements modeling, so in this paper only the sequence diagrams associated to the system Use Cases are considered. Briefly presented, our PDM mainly comprises Petri Nets for the discrete domain of the system and functional blocks for the continuous one. So, we needed to transform very early the UML models into PNs model for two essential goals:

- Formal verification of PIMs based on the properties analysis of the PNs models
- Transformation of PIMS into PSMs with the aim to complete the formal verification by means of the simulation. The simulation is based on a VHDL-AMS heart. So, another model transformation, PNs into VHDL-AMS, has been worked out (Albert et al. 2005).

Since 2002, the OMG initiated a standardization process by issuing a Request for Proposal on Query / Views / Transformations "QVT". This process will lead to an OMG standard for defining model transformations, which will be of interest not only for PIM-to-PSM transformations, but also for defining views on models and synchronization between models. Driven by practical needs and the OMG's request, a large number of approaches to model transformation have recently been proposed. But they mainly are software oriented. In this paper, we propose to transform UML SD-to-PNs whatever the system type is, e.g. for heterogeneous systems. Some papers suggested the Petri net formalism for the modeling of the objects interaction (Paludetto et al. 2004). They proposed transformation rules at model level. We propose to bring the SD-to-PNs transformation at the meta-model level for only a part of the MOF. The reason why will be explained in section two here after.

This paper is made up with six sections. Thus, after this introductive first Section, Section two gives an overview of the context, foundation and limitation of the work. Some SD to PNs rules transformations are presented in Section three. Section four describes both source and target meta-models suited to the transformation. Section five proceeds to the design of the rules transformation meta-model. Finally, the approach retained is assessed and future prospects are outlined in the concluding Section.

2. CONTEXT AND FOUNDATION OF THE WORK

In the translation of sequence diagrams into PNs, the difficulty arises within the framework of the model transformations. Several efforts have addressed transformations at the model level of UML with Petri nets (King and Pooley 2000), (Paludetto et al. 2004), but few have used meta-modeling. However the interest of the meta-modeling is well known now (Ferber and Gutknecht 1998), (Artikson 1997), (Trowitzsch et al. 2005). In this way it is possible to describe and classify the various concepts of a language or a model, and make easier the rules specification of the transformation. This also allows the mapping between the concepts of both meta-models source and target. For a given model, several meta-models may be defined. The suited one depends on the end purpose. Our objective focuses on the requirements V&V, a part based on formal properties and the rest by means of the simulation through VHDL-AMS language. All the necessary tools are layered in a platform according to the MDA approach (see Fig. 1).

This paper concerns the first transformation shown in the Fig. 1. As we said above it is based on a meta-modeling approach. The Fig. 2 models all the transformation process implemented in the platform. Only the first part (dimmed part of the Fig. 2) is discussed in this paper. As shown, the model level has been used only for checking off and to classify the transformation rules. Then, knowing the end purpose of the transformation, the incoming meta-models (SD and PNs) and the transformation meta-model itself may be built and implemented. The next sections will present the approach and the construction of these three meta-models.

When working with UML meta-model, people often refers to the OMG MOF because of its standard sight. This gives an ability to cover all language characteristics for various application domains. Due to its genericity meaning, the MOF is interesting for tool specifications but really heavy for modeling, specifically when the system dynamic must be described formally. In this work an amount of details given by the MOF are not all attractive for the objectives of such a modeling, especially for the requirements modeling. So, we refer to the MOF only for the interesting semantics of the SD at the high level modeling, saying message passing, method activation and data transfer. Indeed, the accordance with the whole MOF standard will be done later, after the validation of the approach.

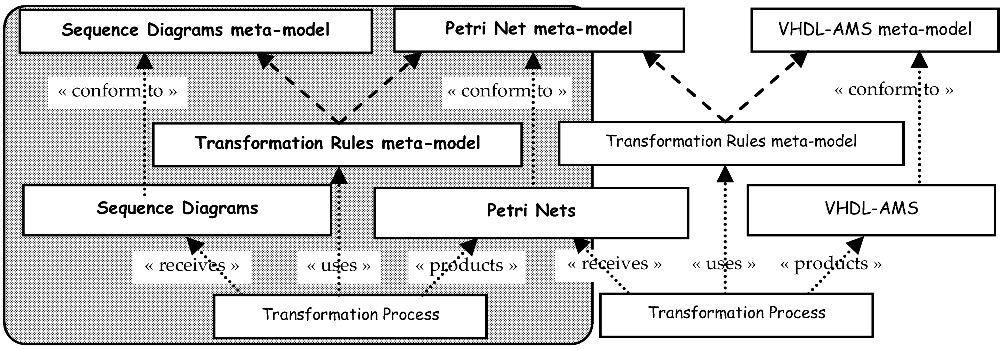


Fig.2. Meta-model based transformation architecture

3. SD INTO PN TRANSFORMATION TECHNIQUES: MODEL LEVEL

As previously mentioned, inter-object communication as well as method and abstract data give a high level view of the system requirements. So, in such a context, sequence diagrams into Petri nets transformation primarily depends on the type of the messages that link up the objects to each other. For instance, a message can be used to express a data transmission or a method call either with or without data transmission. In the same way, a message can wait or not either a result or an acknowledgement. This information can be known if the message semantics is written in accordance with the OCL language (Warmer and Kleppe 2003). This leads to translate synchronous or asynchronous communications into Petri nets.

Specifically we focus on inter-objects communications, as they are described at sub-system levels, and this section assumes either the following interpretations of an UML message: Asynchronous communication, Highly synchronous communication and Loosely synchronous communication.

3.1. Asynchronous communication

Fig. 3 shows a SD asynchronous message and its translation into an asynchronous communication. With Petri net model, such a communication is made up by means of a shared place that is seeing as an outcome place from the sender object (*object1*) and an income place from the receiver object (*object2*) ; the sender (*object1*) and the receiver (*object2*) are represented each one as P-T-P (Place-Transition-Place) Petri net sequence. The different states of the message passing can be the following:

- From the sender point of view, a token on a first P of its P-T-P sequence means the beginning of the sending, the firing of the transition means the sending itself and a token in the second P, the end of the sending.
- From the receiver point of view, a token in the first P of its P-T-P sequence means that it is waiting for a message (token) on a shared place, the firing of the transition means the receiving itself and a token in the second P, the end of the receiving as well as the method running ("op").
- From the message itself, the beginning of the transaction takes place when the sender T is fired and the end when the receiver T is fired in turn.

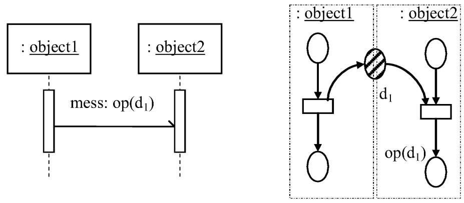


Fig. 3. Mapping of an asynchronous communication

From the system view, the sender puts a token in a shared place (d1) in order to require a method ("op", provided by the receiver), and the receiver accepts the call when its state allows it to consume the d1 token and, early following runs the called method "op".

3.2. Synchronous communication

Highly synchronous message is equivalent to the well known remote procedure call protocol (Fig. 4). The Petri net modeling of such a message is derived from the previous one where the PNs sequence of the sender as well the receiver is built with two merged P-T-P sequences in a P-T-P-T-P resulting sequence. Moreover, two shared places are implemented, one for the call and the second for the return. The centric P of the P-T-P-T-P sequence plays the part of waiting place for the sender and provided method place for the receiver. The second shared place is equivalent to the acknowledge return or result.

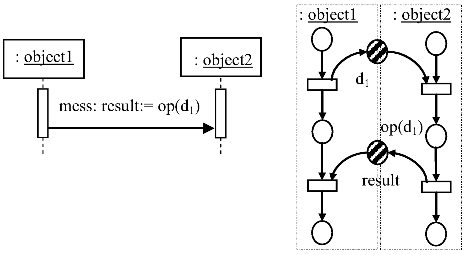


Fig. 4. Highly synchronous communication

The loosely synchronous message is like the highly one with a little bit difference. The sender does not need the result of the operation carried out by the receiver (Fig. 5); it just needs the acknowledgement of the request, and then it continues its own thread of control while the receiver performs its thread too that begin with the provided method ("op (d1)").

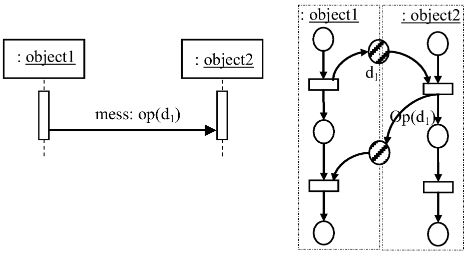


Fig. 5. Loosely synchronous communication

Note that the message syntax of the SD is the same for both asynchronous communication and loosely synchronous communication. In this case, the designer is responsible of the choice of the communication type that will be used in the design carrying on.

4. DEFINITION OF TARGET AND SOURCE META-MODELS

4.1. "Petri nets" meta-model

The objective of this work is to make easier the requirements validation process (see Fig. 1). For the best coherence of the whole transformation process, we used the Petri nets meta-model developed by (Albert et al. 2005) (Fig. 2). However, the working of this meta-model was limited to the representation of inter-objects communications. We extended the Petri nets meta-model in a way shown by Fig. 6.

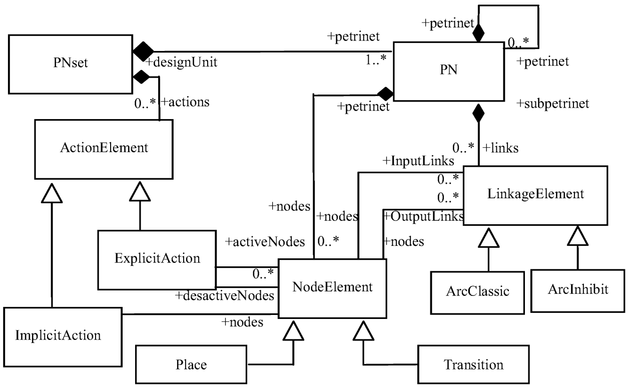


Fig. 6. PN meta-model for SD to PN transformation

As shown in Fig.6, *ArcClassic* and *ArcInhibit* classes both inherit from the *LinkageElement* class; only *ArcClassic* is considered in this paper. The same applies for *ExplicitAction* and *ImplicitAction* that inherits from the *ActionElement* superclass, and are associated to the *NodeElement*: only the second one will be considered in this paper (their activation modes are different: *ExplicitAction* is activated on positive/negative going edge of the associated node, *ImplicitAction* is simultaneously activated with the associated node).

4.2. "Sequence diagrams" meta-model

The objective of the sequence diagrams meta-model is the description of its model elements for their translation into Petri nets. The chosen structure (Fig. 7) describes the elements involved in object interactions.

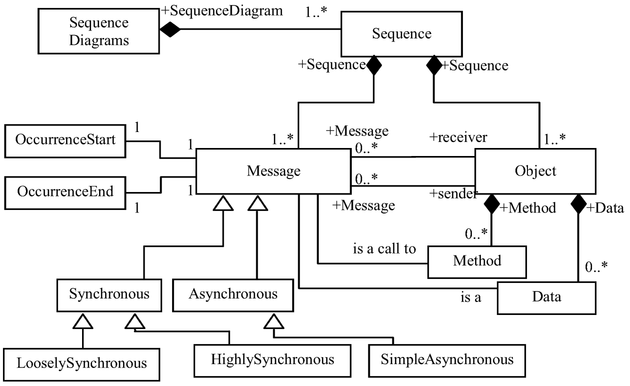


Fig.7. SD meta-model for the transformation of SD into PN

The inter-objects messages can be recognized in this structure as well as their type characterization, synchronous or asynchronous. They express a method call with data exchange. For later use, the edges triggering for start and end message (*OccurrenceStart* and *OccurrenceEnd* classes) are foreseen in order to take into account timed requirements of the communications. For instance, they will be useful for the modeling of the propagation delay of any communication. These timed requirements will not be considered in the transformation presented in this paper.

5. TRANSFORMATION RULES META-MODEL

5.1. The Expression of the Transformation Rules

The transformation rules meta-model (Fig. 8) describes the interactions that exist between classes of the "sequences diagram" meta-model (on the left part of the figure) and "Petri nets" meta-model (right part).

In this transformation, the information of method call with data transfer is associated with a shared place (or communication place *Pcom*) (see *d1* in Fig. 3), whereas the call of method is translated into *ExplicitAction* in relation to receiver description (see *op (d1)* in Fig. 3).

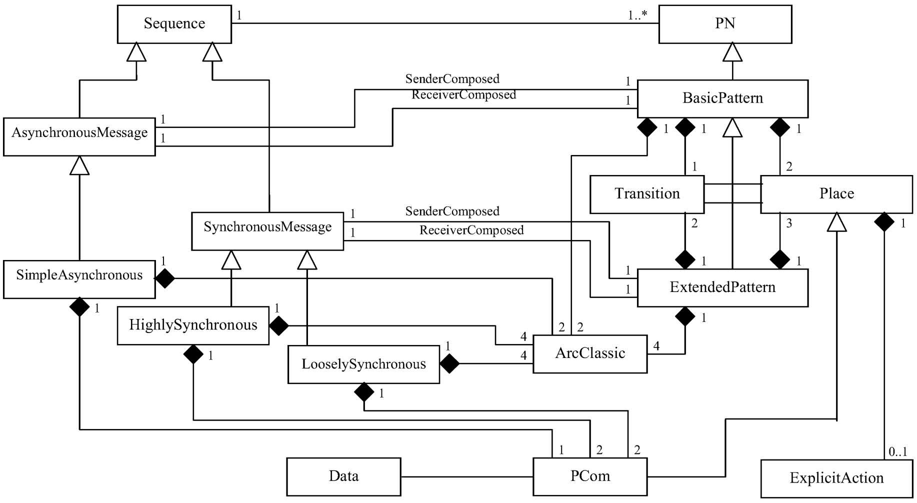


Fig.8. Transformation rules meta-model

In the same way, the result returned when method call is performed is a data associated to a shared place (see *result* in Fig. 4). To express these transformations we retained two Petri net patterns: a *basic* pattern, built with a sequence P-T-P (see *object1* of Petri net in Fig. 3) and an *extended* pattern for P-T-P-T-P sequence (see *objetc1* in Fig. 4). These patterns are in relation with the type of message to be translated, either "AsynchronousMessage" or "SynchronousMessage".

The meta-model distinguishes *Place* and *PCom* classes in order to clarify the transformation rules. Indeed, the places set of the Petri net that results of the message transformation contains the shared places. These later have a specific role: they play the part of an interface between the two communicating objects.

The classes of the transformation rules meta-model are characterized by the provided methods and attributes.

5.2. Execution of the transformation rules

The transformation rules are represented by the meta-model shown in section 5.1; it is important to explain the mechanism used for the transformation of the sequence diagrams into Petri nets.

Let us consider the transformation of an asynchronous message of the SD into its equivalent representation with Petri nets. The main steps of this transformation are: 1) create two single-patterns P-T-P for the two communicating objects (sender and receiver); 2) create the shared place and 3) connect the T sender to the shared place and the shared place to the T receiver. The transformation of the highly synchronous message and the loosely synchronous message into PN is done in similar way.

The transformation mechanism from a source model (SD) to a target model (PN) can be automated with an implementation process based on the Eclipse development platform (http://www.eclipse.org/). The benefit of such a platform is to provide an extensible, universal and versatile integrated development environment. It is possible to create development projects using a given language if this later is "connected" to the platform.

CONCLUSION

In this paper, we presented the meta-modeling based transformation of Sequences Diagrams into Petri nets in order to obtain a model available for the requirements validation. We saw that the meta-model approach opens the way towards a formal transformation and the result, saying PNs, allows the verification of system requirements, in our application. The approach is also interesting for to automate the transformation and to design tools in accordance.

However, many points remain to be deepened and need further investigation. For instance, time does not appear explicitly, but implicitly by the way of order relation. In future work we will consider the explicit time with an aim of taking into account the asynchronous communications with propagation delay. Then, it will be suitable to use the timed Petri net model and thus to enrich the proposed meta-models.

For now, we considered only one sequence diagram. One would think that it is reasonable for the highest level of abstraction, since this sequence diagram may represent the main scenario. But, when considering complex system it is necessary to transform several sequence diagrams using a successive refinement method within an incremental approach: so it is essential to consider the behavior and the interactions of implied objects in several use cases. Such an approach will imply PNs composition.

For the systems of our interest, the use cases generally correspond to operating modes: an object has often several operating modes. The description of the global behavior of the object will be carried out by modes composition (scenarios), after the separate validation of the different modes concerned by this object. The validation of all the modes for a given object will be done by the validation of the resulting Petri net. This task is made easier by the simplicity of the Petri nets patterns implied in the composition (well-structured blocks).

REFERENCES

Albert, V., N'ketsa, A. and Pascal, JC.: "Towards a meta-model based approach for hierarchical Petri net transformations to VHDL". *2005 European Simulation and Modeling Conference*, Porto (Portugal), October 24-26, 2005, pp.531-536

Artikson, C.: "Meta-Modeling for distributed Object Environments". *IEEE Enterprise Distributed Object Computing Workshop*, Oct. 24-26, 1997, pages 90-101, Gold Coast, QLD.

Czarnecki, K. and Helsen, S.: "Classification of Model Transformation Approaches", *OOPSLA'03 Workshop on Generative Techniques in the Context of Model-Driven Architecture*

Esteban, P., Ouardani, A., Paludetto, M. and Pascal, JC.: "A component based approach for system design and virtual prototyping". *12th Annual European Concurrent Engineering Conference* (ECEC'2005), Toulouse (France), April 11-13, 2005, pp.85-90.

Ferber, J. and Gutknecht, O.: "A meta-model for the analysis and design of organizations in multi-agent systems". *IEEE Multi Agent Systems*, July 3-7, 1998, pages. 128-135, Paris.

Frankel, D.: *Model Driven Architecture: Applying MDA to Enterprise*. Computing Wiley Press, 2003

Gerber, A., Lawley, M., Raymond, K., Steel, J. and Wood, A.: "Graph Transformation", *First International Conference (ICGT 2002)*, Barcelona, Spain, October 7-12, 2002. Proceedings. LNCS vol. 2505, Springer-Verlag, 2002, pp. 90 - 105

King, P. and Pooley, R.: "Derivation of Petri Net Performance Models from UML Specifications of Communications Software. Computer Performance Evaluation, Modeling Techniques and Tools", *11th International Conference, TOOLS 2000*, Schaumburg, IL, USA, March 2000. Proceedings / Boudewijn R. Haverkort, Henrik C. Bohnenkamp, Connie U. Smith (Eds.) -Springer Verlag, 2000, pp. 262-276.

Kleppe, A., Warmer, J. and Bast, W.: *MDA explained: The Model Driven Architecture, Practice and Promise*. Addison Wesley, 2003

Paludetto, M., Delatour, J. and Benzina, A.: "UML et réseaux de Petri - Vers une formalisation des besoins des systèmes embarqués". *Technique et science informatiques* "TSI", page n° 543 Fascicule N° 4, Vol N°23, 2004.

Trowitzsch, J., Zimmermann, A. and Homme, G.: "Towards Quantitative Analysis of Real-Time UML Using Stochastic Petri Nets". *19th Parallel and distributed processing symposium IPDPS'05* "IEEE", April 04-08, 2005.

Warmer, J. and Kleppe, A.: *The Object Constraint Language Second Edition: Getting your Models for MDA*. Addison Wesley, 2003.

STATE CLASS GRAPH FOR FUZZY TIME PETRI NETS

J. Cardoso[1], Xiaoyu Mao[2] and Robert Valette[3]
[1] IRIT-UT1, 21, allée de Brienne, 31042 Toulouse France
[2] Almende / University of Maastricht, The Netherlands
[3] LAAS-CNRS, 31077 Toulouse, France
e-mail: jcardoso@univ-tlse1.fr, xmao@almende.com, robert@laas.fr

KEYWORDS
Petri nets, time Petri nets, analysis, fuzzy Petri nets

ABSTRACT

The objective of this paper is the formal verification of quantitative temporal properties of embedded systems. Our approach is different from the current state of the art where the verification of the properties is done after a complete design with numerical values for all parameters. The application domain is that of concurrent engineering design, where essential parameters may be ill-known. The formal mathematical framework used to take into account these ill-known constraints is the fuzzy set theory. The use of fuzzy theory for temporal constraints and Petri net theory for the dynamic aspect in the construction of a graph of states classes allows ensuring some properties during design phases where the value of some parameters is not well defined.

INTRODUCTION

Property verification is typically based on model checking. When continuous time is involved, the number of states is infinite and it is necessary to cover them by means of a finite number of state classes.

We presented in (Mao 05) the tool GraphC that generates, for t-Time Petri Nets where temporal intervals are associated with transitions (see figure 1.a), a graph of classes as the one in figure 5. The state space exactly (in a quantitative way) defines the set of constraints that have to be verified by the transition firings (event occurrences in the real system). The graph of classes is constructed in such a way that the constraints which have to be verified by the occurrence dates for any event sequence in the real system are directly derived by concatenating the constraints attached to the arcs covered by the corresponding sequence in the graph of classes. Each path in this graph is associated with a sequence which can effectively be fired in the Petri net and which verifies the temporal constraints. It must be underlined that this approach allows the analysis of properties such that, for example, “a state is reachable”. But above all, it allows determining the set of temporal constraints the sequence leading to this state must meet.

The problem with using only imprecise temporal parameters delimited by intervals [a b], is that we can only say Yes or No for a property verification. We would like to say not only *Yes* or *No* but also *”It is Possible with a possibility measure”* i.e. with a *quantitative* view.

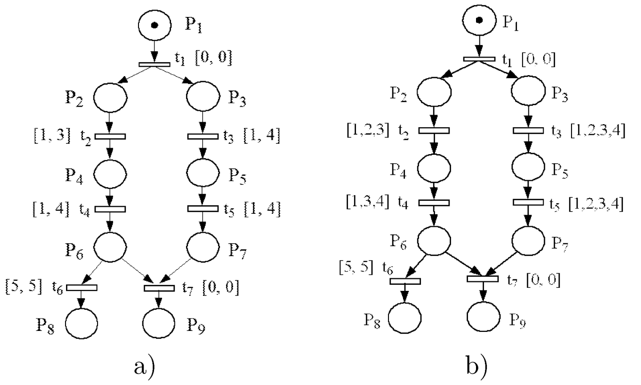


Figure 1: a) t-time and b) fuzzy time Petri nets

This kind of information can be obtained by using a Fuzzy Time Petri Net (Cardoso 98) like the one represented in the figure 1.b, where a temporal constraint is represented by a fuzzy interval of time – representing a possibility distribution of a firing date associated with each transition. For example, the fuzzy interval associated with t_3 is [1 2 3 4]. It means that all instants of time between the *core* [2 3] are the best candidates to the firing interval of t_3, but the ones between the *support* [1 4] are not excluded.

Possibility theory is a very straightforward theory to manage incomplete information; it is particulary useful when there is no available statistical data. In this case, models requiring statistical data, as Stochastic Petri nets or Markov chains, become meaningless. It is more interesting to deal with available data (even if ill-known) than trying to use information that does not exist. In fact, if some verification can be done with values that are ill known at an earlier design phase, it allows refining temporal constraints and providing in this way flexibility in the achievement of goals in the system.

For example, let us consider reconfiguration policies concerning computer and buses. Such systems are used to control critical equipments in planes or cars. The correctness of their design depends on the duration of the cycle time of each equipment. Proving the properties in the proposed graph, obtained from a fuzzy time

Petri net, allows delaying the moment at which a precise (or less imprecise) quantitative value is assigned to some parameters.

A first attempt to do this is (Cardoso 05), where the generated graph of a Fuzzy time Petri Net was an extension of the graph of classes in linear mode (Berthomieu 04) that allows LTL property model checking. The first difference in relation to this work is the kind of the abstract class. The one used in this work is based on (Mao 05) where each path in the state space is associated with a sequence effectively firable in the Petri net. The second difference is that in (Cardoso 05) the graph was generated directly by calculating the Possibility and Necessity measures of the transition firing using the fuzzy intervals associated with the transitions. This implies intersections and subtractions between fuzzy sets, and sometimes the resulting fuzzy sets are not trapezoids and it is necessary to approximate them by trapezoids in order to have a feasible calculus.

The approach proposed in this paper consists in transforming a Fuzzy time Petri Net into several t-Time Petri Nets using the concept of α-cut. In fact, a fuzzy set can be considered as a collection of nested (classical) sets called α-cuts. For each t-Time Petri Net a graph of classes is generated using the tool GraphC. The issue is then to compose all theses graphs into a Fuzzy Graph of Classes, using an algorithm presented in the paper.

The paper is organised as follows: the first Section introduces the basic concepts needed in the proposed approach, as fuzzy dates, α-cut and fuzzy Petri nets. The second Section presents a brief description of GraphC and introduces an illustrative example. The third Section explains how the GraphC and the fuzzy intervals are combined into a Fuzzy GraphC and the advantage of using alpha cuts. Finally, the last Section presents some conclusions and future work.

BASIC NOTIONS

In this section the main notions used in the proposed approach are presented.

Fuzzy date

A date a has *only one value*, which may be ill-known and the fuzzy set of its possible values is a disjunctive set (Yager 84). The available knowledge about a date a is represented by a possibility distribution $\pi_a(\tau)$, $\tau \in \mathcal{T}$ (the universal set of time instants), delimited by the fuzzy set A, represented by the quadruple $[\underline{a}, a_*, a^*, \overline{a}]$ (Dubois 89) which respresents a trapezoid.

The greater $\pi_a(\tau)$, the greater the possibility that a is equal to τ. In the time interval $[\underline{a}, \overline{a}]$ (called support), $0 < \pi_a \leq 1$. In the time interval $[a_*, a^*]$ (called core), $\pi_a(\tau) = 1$. Outside the interval $[\underline{a}, \overline{a}]$, $\pi_a(\tau) = 0$. The wider the support of the fuzzy set, the higher the uncertainty. There are three particular cases: a) triangular form, $a_* = a^* = a$, denoted by $[\underline{a}, a, \overline{a}]$; b) imprecise case, $\underline{a} = a_*$ and $a^* = \overline{a}$, denoted by $[\underline{a}, \overline{a}]$; c) precise case, $\underline{a} = a_* = a^* = \overline{a}$.

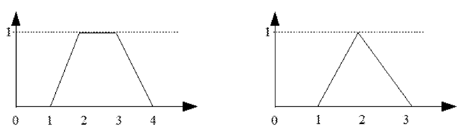


Figure 2: Fuzzy dates : a) a trapezoid ; b) a triangle.

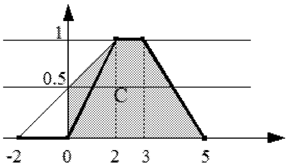


Figure 3: A non trapezoidal fuzzy set

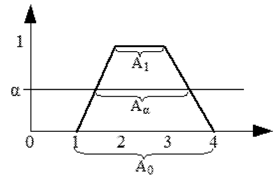


Figure 4: Example of an α-cut

Let us consider the example depicted by figure 1.b. Transitions t_3, t_4 and t_5 are associated with [1 2 3 4] (fig. 2.a) and transition t_2 is associated with the fuzzy temporal interval [1 2 3] (fig. 2.b). In the latest case, the values of the support [1 3] correspond to possible values and the ones of the core [2 2] correspond to the normal behaviour. The intervals [0 0] and [5 5] correspond to precise firing dates.

All operations (addition, subtraction, union, intersection, complement) on fuzzy sets can be done using the extension principle in an easy way if the fuzzy sets are trapezoids or triangles. Otherwise, approximations are required in order to have feasible calculus. The difficulty of such calculus results from the fact that the intersection of two trapezoids is not necessarily a trapezoid.

Let us consider the fuzzy set $[-2\ 2\ 3\ 5]$ of figure 3.a, and let us assume that it is a quantitative constraint between the firing of two transitions which have to be ordered in a sequence. It is thus necessary to compute its intersection with the interval $[0\ \infty)$. The fuzzy set C represented by points $(0, 0), (0, 0.5), (2, 1), (3, 1)$ and $(5, 0)$ is obtained and it is not a trapezoid.

It is possible to avoid this problem and consequently to avoid approximating non trapezoid fuzzy sets by trapezoid fuzzy sets by using the concept of alpha-cut (Dubois 88). In this paper we will proceed in this way.

Alpha cut of a fuzzy set

Definition 1 *Given a fuzzy set A, the alpha-cut (or α-cut) set of A is defined by* $A_\alpha = \{\tau \mid \mu_A(\tau) \geq \alpha\}, \alpha \in]0, 1]$.

As the set A_α is a crisp set, the alpha-cut sets are a mean to convert a fuzzy set into a collection of crisp sets, "facilitating" the computations. The set A_1 is obtained for the core, with level $\alpha = 1$ and the set A_0 is obtained for the support, with level $\alpha = \epsilon > 0$ (see fig. 4).

Property 1 *The α-cut of A are nested sets of $\mathcal{T}$: if $\alpha_1 \geq \alpha$, then $A_{\alpha_1} \subseteq A_\alpha$. The level $\alpha = 1$ corresponds to the core, and $\alpha = 0$ to the universal set.*

For all fuzzy sets A and B of $\mathcal{T}$, and for all α-cuts of]0 1], it is equivalent to directly carry out fuzzy operations on A and B, and then construct the α-cut, or construct first the α-cut and then to carry out on these α-cut the corresponding classical operations. So we have:
- $(A \cup B)_\alpha = A_\alpha \cup B_\alpha$
- $(A \cap B)_\alpha = A_\alpha \cap B_\alpha$
- if $A \subseteq B$ then $A_\alpha \subseteq B_\alpha$

Property 2 *A fuzzy set can be represented by its α-cuts. If the nested α-cuts of a unknown fuzzy set A are known for all thresholds α, the membership function of A can be constructed considering the thresholds from the largest to the smallest.*

In the approach presented in this paper, this property allows to rebuild the fuzzy constraints for the state classes and the analysed scenarios.

Fuzzy Time Petri nets

A t-time Petri net (Merlin 74) allows taking watchdogs into account in a natural way. In this model, an interval $[a_i, b_i]$ is associated with each transition t_i and can be considered as a way of defining an *imprecise* enabling duration.
In a Time Fuzzy Petri net (TFPN) (Cardoso 98) the imprecise enabling duration $[a_i, b_i]$ is extended to a fuzzy duration defined by a possibility distribution.

Definition 2 *A Fuzzy Time Petri Net (FTPN) is a 3-tuple $< \mathcal{N}, M_0, I >$ where:*

- *$\mathcal{N} =< P, T, Pre, Post >$ is a Petri net,*
- *M_0 : is the initial marking,*
- *$I : T \rightarrow (Q^+ \cup 0) * (Q^+ \cup \infty) * (Q^+ \cup \infty) * (Q^+ \cup \infty)$ is the static interval function represented by a fuzzy set.*

The fuzzy static interval function I associates with each transition t_i a temporal interval $[\underline{a_i}, a_{i*}, a_i^*, \overline{a}_i]$ (see the FTPN of figure 1.b and fuzzy sets of figure 2) that represents the set of its possible firing dates counting from its enabling date.

Simple temporal network (STN)

Definition 3 (Simple temporal network) *A simple temporal network (STN) (Dechter 91) N is composed of a finite set V of variables v_i and a finite set C of* **binary** *constraints $C_{ij}(v_i, v_j)$ defined as convex intervals $[c_{mij}, c_{Mij}]$ delimiting the possible distance between two variables v_i and v_j of V. Each C_{ij} is therefore equivalent to: $c_{mij} \leq v_j - v_i \leq c_{Mij}$ $v_i, v_j \in V$.*
A STN is complete if a constraint C_{ij} is associated with each pair of variables v_i and v_j. If no constraints is defined, it can be assumed that $C_{ij}(v_i, v_j) = (-\infty + \infty)$ for the sake of completeness.

Figure 6.a depicts an STN with $V = \{x_1, x_2, x_3\}$ and constraints $C_{12} = C_{13} = [1\ 3]$ and $C_{32} = [0\ 2]$.

Definition 4 (STN inclusion) *A STN $N^1 = (V^1, C^1)$ is included (or nested) in a STN $N^2 = (V^2, C^2)$, noted $N^1 \subseteq N^2$ if and only if:*

- *they are assumed to be complete;*
- *they are based on the same set of variables: $V^1 = V^2 = V$;*
- *$\forall v_i, v_j \in V$, $C^1_{i,j}(v_i, v_j) \subseteq C^2_{i,j}(v_i, v_j)$, i.e. the constraints interval between the arcs of N^1 are included (or nested) into the constraints interval of the corresponding arcs of N^2.*

Definition 5 (A fuzzy STN) *A fuzzy Simple Temporal Network (STN) $N_F = (V, C_F)$ is a STN where the constraints are expressed by means of fuzzy intervals.*
Each alpha-cut (the same alpha level for all the constraints) is therefore a classical STN.

THE TOOL GRAPHC

The tool GraphC generates a graph of classes proposed in (Mao 05) that allows obtaining the exact temporal constraints for a sequence. The reader should refer to (Mao 05) for a detailed description of its construction. In this section only the main lines are given.
Typically a state graph class of a Petri net is constructed in order to check some property expressed in LTL or CTL. In the case of the tool GraphC the objective is to exactly characterise the temporal constraints which have to be verified by the transition firings in order to implement some scenario i.e. some sequence. It is why sets of constraints are associated with the arcs, and not only with the classes. The second difference (a consequence of the first one) is that in place of a set of clocks and inequality matrices, the temporal constraints are expressed under the form of Simple Temporal Networks (STN) which are complete and minimal (Floyd-Warshall algorithm).
Before presenting the definitions of a class and of the set of constraints associated with an arc between two classes, let us introduce an example in order to illustrate the graph generated by GraphC and later, by the Fuzzy GraphC.
Let us consider the Petri net of figure 1.a, as the representation of a main process that call a distant process. The main process is described by transitions t_1, t_2, t_4, t_6 and t_7. The distant process is described by transitions

t_3 and t_5. Transition t_7 corresponds to the reception of the response (a token in place p_9) after the call had been done (a token in p_3).

Several questions can be asked concerning the behaviour of the system modelled by this Petri net. One of them is whether the response arrives in time or not. A watchdog is used to avoid waiting for too long. If there is a failure and the response does not arrive, transition t_6 must be fired. The problem is that the response can arrive too late. The objective is to make a compromise between useless long wait in case of failure and the loss of some very late responses. It is interesting to evaluate different compromises according to the watchdog values associated with t_6.

We want to determine for which firing intervals transition t_6, representing the watchdog, can be fired. In fact, as t_7 is fired immediately after t_5, t_6 is only fired if it is fired before t_5.

Definition 6 *A* **state class** $\mathcal{C}$ *obtained by the firing of transition* t_i *is defined by the pair* $\{M, Nc\}$ *where:*
- M is the (reachable) marking obtained by the firing of t_i*;*
- Nc is a minimal and complete simple temporal network composed of : 1) the variable x_i *associated with the firing of transition* t_i*; 2) for each enabled transition* t_k *from M, the variable associated with the firing of transition* t_l *which has enabled* t_k *; 3) the temporal constraints between these variables.*

Definition 7 *The* ***simple temporal network*** $Nt_{j,c}$ *delimiting the firing of* t_j *from class* $\mathcal{C} = \{M, Nc\}$ *is a complete and minimal simple temporal network (see def. 3) over the following variables and constraints:*
- those of Nc,
- the variable x_j *representing the firing date of* t_j*,*
- the upper bound of the firing intervals for all the enabled transitions in $\mathcal{C}$*.*

The last point of the preceding definition is required in order to guarantee the so-called strong semantics of t-Time Petri Nets (when the upper bound of the firing interval of t is reached one enabled transition - t or another one - has to be fired).

It may happen that a transition t_j can only be fired from some states belonging to a class $\mathcal{C}$ and not from all its states. It is therefore necessary to introduce a restriction of $\mathcal{C}$ which only contains those states from which t_j can be fired.

Definition 8 *The restricted class* $\mathcal{C}r^j = (M_r, Nc_r)$ *of class* $\mathcal{C}$ *is defined by*
- $M_r = M$*,*
- Nc_r *is the network* $Nt_{j,c} \cap Nc$ *after the application of Floyd-Warshall algorithm.*

For each restricted class $\mathcal{C}r^j$ it must be indicated the original (or mother) class $\mathcal{C}$.

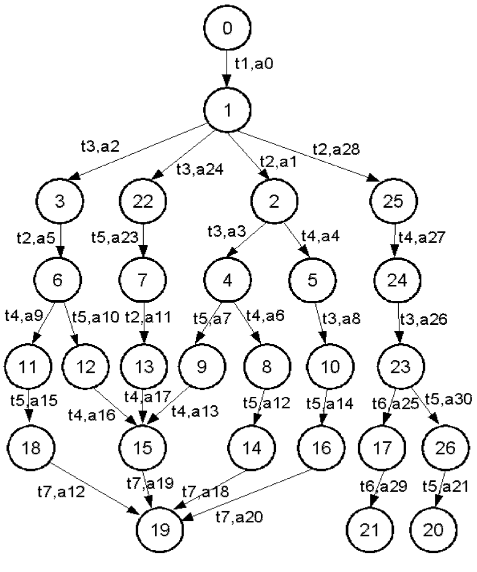


Figure 5: The graph of t-time Petri net of fig. 1.a

Definition 9 *The graph of classes* $G = (\mathcal{N}, \mathcal{A})$ *is composed of the set* $\mathcal{N}$ *of vertices (the state classes)* $\mathcal{C} = (M,\ Nc)$ *and the set* $\mathcal{A}$ *of arcs* $a = (\mathcal{C}, \mathcal{C}')$ *connecting the vertices. An arc a is labelled by a transition t (leading from* $\mathcal{C}$ *to* $\mathcal{C}'$*) and* $Nt_{i,c}$ *delimiting its firing date.*

Let us consider the TPN in figure 1.a. The graph of state classes generated by the tool GraphC is the one in fig. 5. The initial class is given by $\mathcal{C}_0 = (p_1,\ x_0)$ where x_0 represents the beginning of the world; we have also $\mathcal{C}_1 = (p_2p_3,\ x_1)$ and $\mathcal{C}_3 = (p_2p_5,\ (x_1, x_3) = [1\ 3])$. It means that class $\mathcal{C}_3$ gather all states such that the marking is p_2p_5 (one token in p_2 and in p_5) and the temporal constraint is such that the firing of t_3 must occur during the interval [1 3] after that of t_1. The STN of arcs a_0, a_2 and a_5 are, respectively, $Nt_{1,0} : (x_0, x_1) = [0\ 0]$, $Nt_{3,1} : (x_1, x_3) = [1\ 3]$ and $Nt_{2,3}$ as depicted by fig. 6.a. The class $\mathcal{C}_{22} = (p_2p_5,\ (x_1, x_3) = [1\ 2])$ is a restricted class of $\mathcal{C}_3$.

The simple temporal network of a sequence s represents all the temporal constraints between all transition firings in the sequence. It is obtained by the union of the STN of each transition in s. For example, let us consider the sequence $s_1 = t_1; t_3; t_2; t_4; t_5; t_7$ in the TPN of figure 1.a, that corresponds to a path following the arcs a_0, a_2, a_5, a_9, a_{15} and a_{22} in the graph of figure 5. Each transition firing is represented by a STN (see def. 7); the whole sequence s is characterised by its STN, noted $Ns_0 = Nt_{1,0} \cup Nt_{3,1} \cup Nt_{2,3} \cup Nt_{4,6} \cup Nt_{5,11} \cup Nt_{7,18}$ presented in figure 6.b. Each node x_i in Ns_0 corresponds to the firing of transition t_i.

THE FUZZY GRAPHC

The fuzzy GraphC $G_F = (\mathcal{N}_F, \mathcal{A}_F)$ is the extension of the GraphC to the case of Fuzzy Time Petri nets in place of t-Time Petri Nets. This means that the Simple

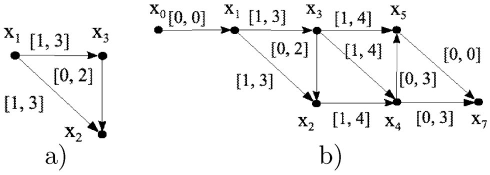


Figure 6: a) The STN $Nt_{2,3}$, b)The STN of s_0

Temporal Networks attached to the state classes and to the arcs become Fuzzy Simple Temporal Networks. Any problem due to non trapezoid fuzzy sets (particularly for the execution of Floyd-Warshall algorithm) is avoided by operating on the graphs generated for each alpha-cut separately.

Decomposing a FTPN into TPN

First of all, all fuzzy time intervals $I(t_i), i = 1, ..., 7$ of the Petri net of fig. 1 are converted into ten α-sets, $I_\alpha(t_i)$. Each set $I_\alpha(t_i), i = 1, ..., 7$ represents a t-time Petri net (since the time intervals are imprecise or crisp), noted α-Petri net for short.
The 0-Petri net of fig. 1.a is the graphical representation for $\alpha = 0$ (the support). For example, the time intervals for $\alpha = 1, 0.9, 0.5$ and 0.1 are:

- for $\alpha = 1$ (the core): $I_1(t_1) = I_1(t_7) = [0\ 0], I_1(t_2) = [2\ 2], I_1(t_3) = I_1(t_5) = [2\ 3], I_1(t_4) = [3\ 3]$, $I_1(t_6) = [5.\ 5.]$.
- for $\alpha = 0.9$: $I_{.9}(t_1) = I_{.9}(t_7) = [0\ 0], I_{.9}(t_2) = [1.9\ 2.1], I_{.9}(t_3) = I_{.9}(t_5) = [1.9\ 3.1], I_{.9}(t_4) = [2.8\ 3.1]$ and $I_{.9}(t_6) = [5.\ 5.]$.
- for $\alpha = 0.5$: $I_{.5}(t_1) = I_{.5}(t_7) = [0\ 0], I_{.5}(t_2) = [1.5\ 2.5], I_{.5}(t_3) = I_{.5}(t_5) = [1.5\ 3.5], I_{.5}(t_4) = [2.0\ 3.5]$ and $I_{.5}(t_6) = [5.\ 5.]$.
- for $\alpha = 0.1$: $I_{.1}(t_1) = I_{.1}(t_7) = [0\ 0], I_{.1}(t_2) = [1.1\ 2.9], I_{.1}(t_3) = I_{.1}(t_5) = [1.1\ 3.9], I_{.1}(t_4) = [1.2\ 3.9]$ and $I_{.1}(t_6) = [5.\ 5.]$.

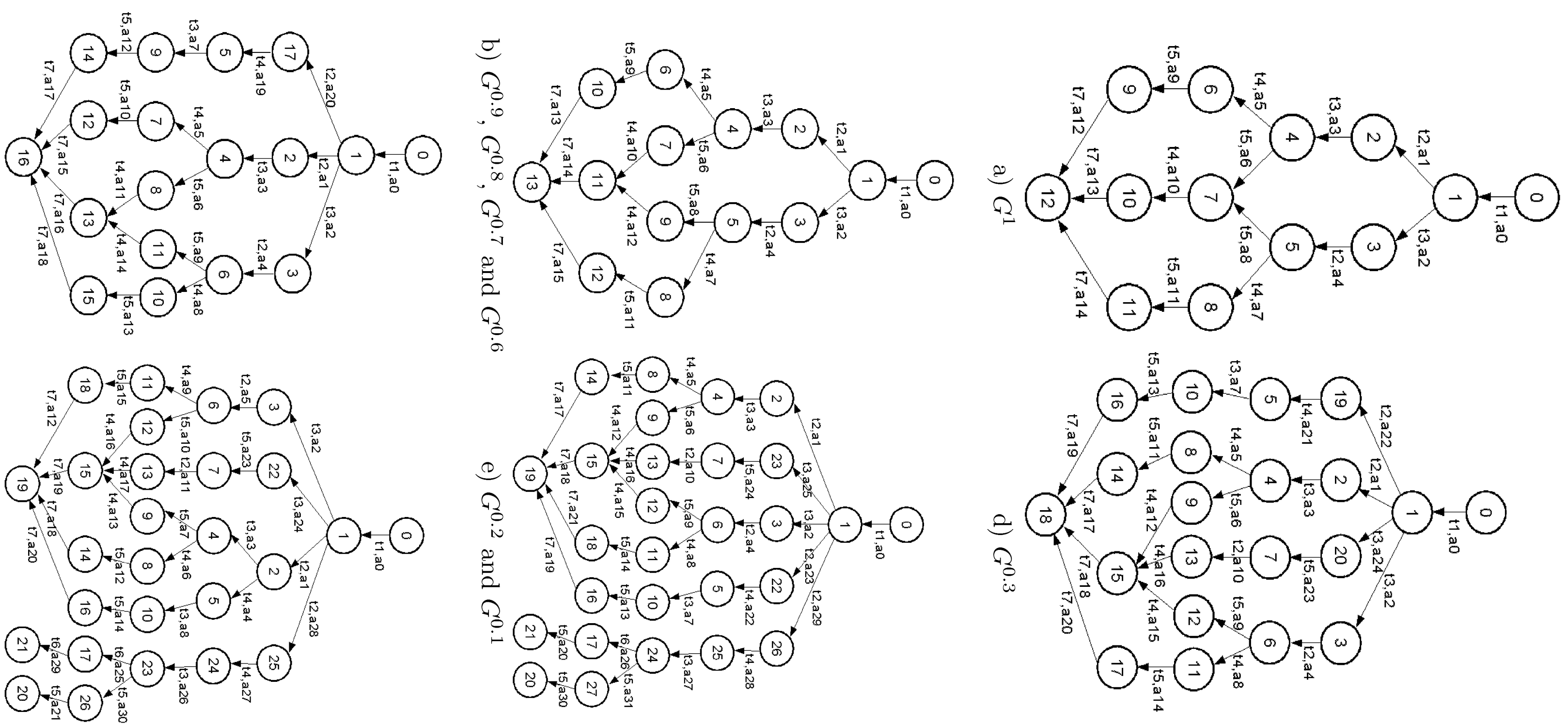
According to Property 1, for all α sets $I_\alpha(t)$ for a transition t_i, they are nested: $I_1(t_i) \subseteq I_{.9}(t_i) \ldots \subseteq I_{.1}(t_i) \subseteq I_0(t_i)$.
The following notations are used in the remainder of the paper: $G^\alpha = (\mathcal{N}^\alpha, \mathcal{A}^\alpha)$ is a graph generated by an α-Petri net; $[a]^{\alpha_1}$ an arc of a graph G^{α_1}, $[\mathcal{C}]^{\alpha_1}$ a class of a graph G^{α_1}, and s^{α_1} a sequence of a graph G^{α_1}. A class $[\mathcal{C}]^{\alpha_1}$ is noted $[(M,\ Nc)]^{\alpha_1}$ or $([M]^{\alpha_1}, [Nc]^{\alpha_1})$. An arc $[a]^{\alpha_1}$ is labelled by a transition t_i and the STN $[Nt_{i,c}]^{\alpha_1}$ delimiting this firing.
For each α-Petri net, a state class graph G^α is constructed; fig. 5 corresponds to G^0 (support). The other graphs for the other values of α are depicted in figure 7 and all classes for all these graphs are represented in Table 1. The first two columns indicate the marking and the STN constraints (in this example there is just a pair of nodes) for all classes. The following columns indicate the name of the classes for each α-cut, and the value of α. The name of the classes of the composed fuzzy graph G_F is listed in the last column. Each line indicates: the marking M_i, the STN Nc_i, and, for each set of columns the name of the classes in G^α and the constraint value. An empty value in a column labeled by α and a line labelled by $\mathcal{C}_{F_i}$ indicates that there is no class included in $\mathcal{C}_{F_i}$ for this G^α. We can see in figure 7 that some graphs have the same structure (as $G^{0.9}$, $G^{0.8}$, $G^{0.7}$ and $G^{0.6}$); in these case, the name of the class (and arcs) are the same but their STN can be different as it can be seen in Table 1. The list of restricted classes with their "original" (or mother) class (G^1 and $G^{0,9}$ have no restricted classes) is : $[\mathcal{C}_{17}]^{0.5} = [\mathcal{C}_2]_r^{0.5}$, $[\mathcal{C}_{19}]^{0.3} = [\mathcal{C}_2]_r^{0.3}$, $[\mathcal{C}_{20}]^{0.3} = [\mathcal{C}_3]_r^{0.3}$, $[\mathcal{C}_{22}]^{0.2} = [\mathcal{C}_2]_r^{0.2}$, $[\mathcal{C}_{23}]^{0.2} = [\mathcal{C}_3]_r^{0.2}$, $[\mathcal{C}_{24}]^{0.2} = [\mathcal{C}_{10}]_r^{0.2}$, $[\mathcal{C}_{25}]^{0.2} = [\mathcal{C}_5]_r^{0.2}$, $[\mathcal{C}_{26}]^{0.2} = [\mathcal{C}_{22}]_r^{0.2}$, $[\mathcal{C}_{27}]^{0.2} = [\mathcal{C}_{16}]_r^{0.2}$, $[\mathcal{C}_{25}]^0 = [\mathcal{C}_2]_r^0$, $[\mathcal{C}_{22}]^0 = [\mathcal{C}_3]_r^0$, $[\mathcal{C}_{24}]^0 = [\mathcal{C}_5]_r^0$ and $[\mathcal{C}_{26}]^0 = [\mathcal{C}_{16}]_r^0$ (see def. 8).
The next step is to build the fuzzy state class graph G_F from all these graphs G^α as described in the sequel.

Composing several GraphC G^α in a Fuzzy GraphC G_F

Principle
The objective is to derive a Fuzzy GraphC G_F from all G^α for each α-Petri net. In order to do this, it is necessary to match all the G^α in order to find how their classes can be grouped in order to derive the classes of G_F and then reconstruct for each class (and arc) the fuzzy STN representing them. Although the problem of matching two graphs is an NP complete problem, in this case the problem is much simpler because the firing sequences are necessarily preserved.
It follows indeed from the alpha cut definition that $G^{\alpha_1} \subseteq G^\alpha$ for $\alpha_1 \geq \alpha$, the only problem is to find which classes of G^{α_1} are contained in those of G^α, since the structure of the graphs (number of nodes and arcs) can be different due to the restricted classes that can appear or disappear when α changes (no matter if it increases or decreases). In other words, in order to have the same graph structure for all the G^α, it is sometimes necessary for some G^α to introduce a restricted class which is not actually restricted for the current value of α because it exists for some other values of α. Similarly, two classes may be equivalent within one G^α and not within other ones. They have then to contribute for this α-cut to two different classes in the Fuzzy GraphC as if they were not equivalent.

Matching the state class graphs of two α-cuts
The algorithm for finding the corresponding state classes in two state class graphs (G^{α_1} and G^α) in order to build the fuzzy state class graph is the algorithm 1. The function IsIncludedClass is based on definition 10:

Definition 10 (Classes inclusion) *A class* $[\mathcal{C}]^{\alpha_1} = [(M,\ Nc)]^{\alpha_1}$ *(of graph* G^{α_1}*) with* $[Nc = (V, C)]^{\alpha_1}$ *is*



a) G^1

d) $G^{0.3}$

b) $G^{0.9}$, $G^{0.8}$, $G^{0.7}$ and $G^{0.6}$

e) $G^{0.2}$ and $G^{0.1}$

c) $G^{0.5}$ and $G^{0.4}$

f) G^0

Figure 7: Graphs of classes G^α

M_i	Nc_i	$\mathcal{C}_i$	$\alpha = 1$	$\mathcal{C}_i$	$\alpha = 0.9$	$\alpha = 0.8$	$\alpha = 0.7$	$\alpha = 0.6$	$\mathcal{C}_i$	$\alpha = 0.5$	$\alpha = 0.4$	$\mathcal{C}_i$	$\alpha = 0.3$	$\mathcal{C}_i$	$\alpha = 0.2$	$\alpha = 0.1$	$\mathcal{C}_i$	α=0	$\mathcal{C}_{F_i}$
p_1	x_0	$\mathcal{C}_0$		$\mathcal{C}_0$					$\mathcal{C}_0$			$\mathcal{C}_0$		$\mathcal{C}_0$			$\mathcal{C}_0$		$\mathcal{C}_{F_0}$
p_2p_3	x_1	$\mathcal{C}_1$		$\mathcal{C}_1$					$\mathcal{C}_1$			$\mathcal{C}_1$		$\mathcal{C}_1$			$\mathcal{C}_1$		$\mathcal{C}_{F_1}$
p_3p_4	$C_{1,2}$	$\mathcal{C}_2$	[2 2]	$\mathcal{C}_2$	[1.9 2.1]	[1.8 2.2]	[1.7 2.3]	[1.6 2.4]	$\mathcal{C}_2$	[1.5 2.5]	[1.4 2.6]	$\mathcal{C}_2$	[1.3 2.7]	$\mathcal{C}_2$	[1.2 2.8]	[1.1 2.9]	$\mathcal{C}_2*$	[1 3]	$\mathcal{C}_{F_2}$
p_2p_5	$C_{1,3}$	$\mathcal{C}_3$	[2 2]	$\mathcal{C}_3$	[1.9 2.1]	[1.8 2.2]	[1.7 2.3]	[1.6 2.4]	$\mathcal{C}_3$	[1.5 2.5]	[1.4 2.6]	$\mathcal{C}_3$	[1.3 2.7]	$\mathcal{C}_3$	[1.2 2.8]	[1.1 2.9]	$\mathcal{C}_3$	[1 3]	$\mathcal{C}_{F_3}$
p_4p_5	$C_{2,3}$	$\mathcal{C}_4$	[0 1]	$\mathcal{C}_4$	[0 1.2]	[0 1.4]	[0 1.6]	[0 1.8]	$\mathcal{C}_4$	[0 2]	[0 2.2]	$\mathcal{C}_4$	[0 2.4]	$\mathcal{C}_4$	[0 2.6]	[0 2.8]	$\mathcal{C}_4$	[0 3]	$\mathcal{C}_{F_4}$
p_4p_5	$C_{3,2}$	$\mathcal{C}_5$	[0 0]	$\mathcal{C}_5$	[0 0.2]	[0 0.4]	[0 0.6]	[0 0.8]	$\mathcal{C}_6$	[0 1]	[0 1.2]	$\mathcal{C}_6$	[0 1.4]	$\mathcal{C}_6$	[0 1.6]	[0 1.8]	$\mathcal{C}_6$	[0 2]	$\mathcal{C}_{F_5}$
p_5p_6	$C_{3,4}$	$\mathcal{C}_6$	[2 3]	$\mathcal{C}_6$	[1.6 3.1]	[1.2 3.2]	[0.8 3.3]	[0.4 3.4]	$\mathcal{C}_7$	[0 3.5]	[0 3.6]	$\mathcal{C}_8$	[0 3.7]	$\mathcal{C}_8$	[0 3.8]	[0 3.9]	$\mathcal{C}_8$	[0 4]	$\mathcal{C}_{F_6}$
p_4p_7	$C_{2,5}$	$\mathcal{C}_7^+$	[2 3]	$\mathcal{C}_7$	[1.9 3.1]	[1.8 3.2]	[1.7 3.3]	[1.6 3.4]	$\mathcal{C}_8$	[1.5 3.5]	[1.4 3.6]	$\mathcal{C}_9$	[1.3 3.7]	$\mathcal{C}_9$	[1.2 3.8]	[1.1 3.9]	$\mathcal{C}_9$	[1 4]	$\mathcal{C}_{F_7}$
p_5p_6	$C_{3,4}$	$\mathcal{C}_8$	[3 3]	$\mathcal{C}_8$	[2.8 3.1]	[2.6 3.2]	[2.4 3.3]	[2.2 3.4]	$\mathcal{C}_{10}$	[2 3.5]	[1.8 3.6]	$\mathcal{C}_{11}$	[1.6 3.7]	$\mathcal{C}_{11}$	[1.4 3.8]	[1.2 3.9]	$\mathcal{C}_{11}$	[1 4]	$\mathcal{C}_{F_8}$
p_4p_7	$C_{2,5}$	$\mathcal{C}_7^+$	[2 3]	$\mathcal{C}_9$	[1.7 3.1]	[1.4 3.2]	[1.1 3.3]	[0.8 3.4]	$\mathcal{C}_{11}$	[0.5 3.5]	[0.2 3.6]	$\mathcal{C}_{12}$	[0 3.7]	$\mathcal{C}_{12}$	[0 3.8]	[0 3.9]	$\mathcal{C}_{12}$	[0 4]	$\mathcal{C}_{F_9}$
p_6p_7	$C_{4,5}$	$\mathcal{C}_9$	[0 1]	$\mathcal{C}_{10}$	[0 1.5]	[0 2]	[0 2.5]	[0 3]	$\mathcal{C}_{12}$	[0 3.5]	[0 3.6]	$\mathcal{C}_{14}$	[0 3.7]	$\mathcal{C}_{14}$	[0 3.8]	[0 3.9]	$\mathcal{C}_{14}$	[0 4]	$\mathcal{C}_{F_{10}}$
p_6p_7	x_4	$\mathcal{C}_{10}$		$\mathcal{C}_{11}$					$\mathcal{C}_{13}$			$\mathcal{C}_{15}$		$\mathcal{C}_{15}$			$\mathcal{C}_{15}$		$\mathcal{C}_{F_{11}}$
p_6p_7	$C_{4,5}$	$\mathcal{C}_{11}$	[0 0]	$\mathcal{C}_{12}$	[0 0.3]	[0 0.6]	[0 0.9]	[0 1.2]	$\mathcal{C}_{15}$	[0 1.5]	[0 1.8]	$\mathcal{C}_{17}$	[0 2.1]	$\mathcal{C}_{18}$	[0 2.4]	[0 2.7]	$\mathcal{C}_{18}$	[0 3]	$\mathcal{C}_{F_{12}}$
p_9	x_7	$\mathcal{C}_{12}$		$\mathcal{C}_{13}$					$\mathcal{C}_{16}$			$\mathcal{C}_{18}$		$\mathcal{C}_{19}$			$\mathcal{C}_{19}$		$\mathcal{C}_{F_{13}}$
p_3p_6	$C_{1,4}$								$\mathcal{C}_5$	[3.5 3.5]	[3.2 3.6]	$\mathcal{C}_5$	[2.9 3.7]	$\mathcal{C}_5$	[2.6 3.8]	[2.3 3.9]	$\mathcal{C}_5$	[2 4]	$\mathcal{C}_{F_{14}}$
p_5p_6	$C_{4,3}$								$\mathcal{C}_9$	[0 0]	[0 0.4]	$\mathcal{C}_{10}$	[0 0.8]	$\mathcal{C}_{10}$	[0 1.2]	[0 1.6]	$\mathcal{C}_{10}$	[0 2]	$\mathcal{C}_{F_{15}}$
p_6p_7	$C_{4,5}$								$\mathcal{C}_{14}$	[1.5 3.5]	[1.4 4]	$\mathcal{C}_{16}$	[1.3 4.5]	$\mathcal{C}_{16}$	[1.2 5]]	[1.1 5]	$\mathcal{C}_{16}$	[1 5]	$\mathcal{C}_{F_{16}}$
p_3p_4	$C_{1,2}$								$\mathcal{C}_{17}$	[1.5 1.5]	[1.4 1.8]	$\mathcal{C}_{19}$	[1.3 2.1]	$\mathcal{C}_{22}$	[1.2 2.4]	[1.1 2.7]	$\mathcal{C}_2*$	[1 3]	$\mathcal{C}_{F_{17}}$
p_2p_5	$C_{1,3}$											$\mathcal{C}_{20}$	[1.3 1.4]	$\mathcal{C}_{23}$	[1.2 1.6]	[1.1 1.8]	$\mathcal{C}_{22}$	[1 2]	$\mathcal{C}_{F_{18}}$
p_4p_7	x_2											$\mathcal{C}_{13}$		$\mathcal{C}_{13}$			$\mathcal{C}_{13}$		$\mathcal{C}_{F_{19}}$
p_2p_7	$C_{1,5}$											$\mathcal{C}_7$	[2.6 2.7]	$\mathcal{C}_7$	[2.4 2.8]	[2.2 2.9]	$\mathcal{C}_7$	[2 3]	$\mathcal{C}_{F_{20}}$
p_5p_8	$C_{3,6}$													$\mathcal{C}_{17}$	[3.8 3.8]	[3.4 3.9]	$\mathcal{C}_{17}$	[3 4]	$\mathcal{C}_{F_{21}}$
p_7p_8	x_6													$\mathcal{C}_{20}$			$\mathcal{C}_{20}$		$\mathcal{C}_{F_{22}}$
p_7p_8	x_5													$\mathcal{C}_{21}$			$\mathcal{C}_{21}$		$\mathcal{C}_{F_{23}}$
p_5p_6	$C_{4,3}$													$\mathcal{C}_{24}$	[1.2 1.2]	[1.1 1.6]	$\mathcal{C}_{23}$	[1 2]	$\mathcal{C}_{F_{24}}$
p_3p_6	$C_{1,4}$													$\mathcal{C}_{25}$	[2.6 2.6]	[2.3 2.8]	$\mathcal{C}_{24}$	[2 3]	$\mathcal{C}_{F_{25}}$
p_3p_4	$C_{1,2}$													$\mathcal{C}_{26}$	[1.2 1.2]	[1.1 1.6]	$\mathcal{C}_{25}$	[1 2]	$\mathcal{C}_{F_{26}}$
p_6p_7	$C_{4,5}$													$\mathcal{C}_{27}$	[5 5]	[5 5]	$\mathcal{C}_{26}$	[5 5]	$\mathcal{C}_{F_{27}}$

Table 1: Nested classes for all graphs G^α.

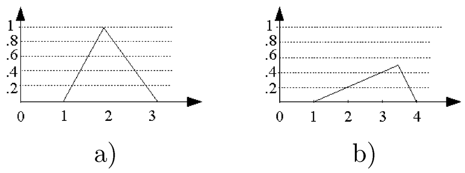


Figure 8: The fuzzy constraint associated with a class: a) $C_{F_{13}}$ of $\mathcal{C}_{F_3}$, b) $C_{F_{14}}$ of $\mathcal{C}_{F_{14}}$

included (or nested) into a class $[\mathcal{C}]^\alpha = [(M,\ Nc)]^\alpha$ (of graph G^α) with $[Nc = (V, C)]^\alpha$, noted $[\mathcal{C}]^{\alpha_1} \subseteq [\mathcal{C}]^\alpha$, if:
- they have the same marking, $[M]^{\alpha_1} = [M]^\alpha$;
- $[Nc]^{\alpha_1}$ is included in $[Nc]^\alpha$ (according def. 4);
- they are reached from the initial state class by the same firing sequences;
- from them the same firing sequences can be fired.

The execution of this algorithm provides a list ListCl with all pairs of classes $([C]^{\alpha_1}, [C]^\alpha)$, $[C]^{\alpha_1} \subseteq [C]^\alpha$ and a list ListArc with all pairs of arcs $([a]^{\alpha_1}, [a]^\alpha)$, $[a]^{\alpha_1} \subseteq [a]^\alpha$. Table 1 was constructed based on ListCl. It can be observed that the constraints for a given class are such that $C_{ij}^{\alpha_1} \subseteq C_{ij}^{\alpha}$ if $\alpha_1 \geq \alpha$ (each class is included in the class at its right side in the same line). A fuzzy class $\mathcal{C}_F$ is "constructed" from all its α cuts (the nested crisp sets) in a same line (see Table 1), using Property 2, as for example, $\mathcal{C}_{F_3}$ whose STN is represented in fig. 8.a. The obtained fuzzy class can be not normalised, for example if there is no class $[\mathcal{C}_i]^1$ (the core) on its line. It is the case of class $\mathcal{C}_{F_{14}}$ represented in fig. 8.b, since the maximum value is $\alpha = 0.5$.
In the example considered here, the structure of the Fuzzy GraphC G_F is that of the GraphC for the α-cuts 0.1 and 0.2 (figure 7.e). This illustrates the fact that it is not necessarily the structure of G^0, the GraphC for the support, that defines the structure of G_F although it is the one describing the less constrained system.

Particular cases of class inclusion
It can be observed in Table 1 that there are two types of class inclusion $[\mathcal{C}]^{\alpha_1} \subseteq [\mathcal{C}]^\alpha$:

1. G^{α_1} and G^α have exactly the same structure. It is the case for: 1) $G^{0.5}$ and $G^{0.4}$, 2) $G^{0.9}$, $G^{0.8}$, $G^{0.7}$ and $G^{0.6}$ and 3) $G^{0.2}$ and $G^{0.1}$.

2. G^{α_1} and G^α do not have the same structure. As mentioned informally above, there are two interesting cases (see Table 1):
   • Case 1: for $\alpha_1 > \alpha$, $[\mathcal{C}_j]^{\alpha_1} \subseteq [\mathcal{C}_k]^\alpha$ and $[\mathcal{C}_j]^{\alpha_1} \subseteq [\mathcal{C}_l]^\alpha$, i.e. a class of G^{α_1} is contained in two classes of G^α. It is the case for G^1 and $G^{0.9}$: $[\mathcal{C}_7^+]^1$ is nested in both classes $[\mathcal{C}_7]^{0.9}$ and $[\mathcal{C}_9]^{0.9}$.

   • Case 2: for $\alpha_1 > \alpha$, $[\mathcal{C}_j]^{\alpha_1} \subseteq [\mathcal{C}_k]^\alpha$ and $[\mathcal{C}_l]^{\alpha_1} \subseteq [\mathcal{C}_k]^\alpha$, i.e. two classes of G^{α_1} are contained in a same class of G^α. It is the case for $G^{0.1}$ and G^0: $[\mathcal{C}_2]^{0.1} \subseteq [\mathcal{C}_2*]^0$ and $[\mathcal{C}_{22}]^{0.1} \subseteq [\mathcal{C}_2*]^0$. By the way, $[\mathcal{C}_{22}]^{0.1}$ is a restricted class of $[\mathcal{C}_2]^{0.1}$.

Algorithm 1 Matching of two graphs (G^{α_1}, G^α), $\alpha_1 \geq \alpha$

```
InclusionGraph($G^{\alpha_1}, G^\alpha$)
nodesIC = add($G^{\alpha_1}$.InitialClass, $\alpha_1$) {Treat the initial class $[C_0]^{\alpha_1}$}
ListCl.add(($G^{\alpha_1}$.InitialClass, $\alpha_1$), ($G^\alpha$.InitialClass, $\alpha$)) {The initial classes are always nested $[C_0]^{\alpha_1} \subseteq [C_0]^\alpha$}
while nonEmpty(nodesIC) do
  $n_1$=nodesIC.get(0){Extract a node from nodesIC ($G^{\alpha_1}$)}
  $n$ = $n_1$.IncludeClass{Treat the corresponding class $[C]^\alpha$, with $[C]^{\alpha_1} \subseteq [C]^\alpha$}
  for i=1 to $n_1$.outputArc.size do
    (arc$_1$, $\alpha_1$)=($n_1$.outputArc.get(i), $\alpha_1$){take an output arc of $[C]^{\alpha_1}$}
    for j=1 to $n$.outputArc.size do
      (arc, $\alpha$)=($n$.outputArc.get(j), $\alpha$){take an output arc of $[C]^\alpha$}
      {If both arcs in $G^{\alpha_1}$ and $G^\alpha$ are labelled by the same transition}
      if arc$_1$.trans=arc.trans then
        {Verify if the target classes are included}
        if IsIncludedClass((arc$_1$.targetCl, $\alpha_1$), (arc.targetCl, $\alpha$)) then
          ListCl.add((arc$_1$.targetCl.getId, $\alpha_1$), (arc.targetCl.getId, $\alpha$)) {add the pair $[C_T]^{\alpha_1} \subseteq [C_T]^\alpha$}
          nodesIC = add(arc$_1$.targetCl) {add this class to be analysed later if it is not yet}
          if IsIncludedSTN((arc$_1$.STN, $\alpha_1$), (arc.STN,$\alpha$)) then
            ListArc.add((arc$_1$.getId, $\alpha_1$), (arc.getId, $\alpha$)) {add the pair $[\text{arc}_1]^{\alpha_1} \subseteq [\text{arc}]^\alpha$}
          end if
        end if
      end if
    end for
    ListArc=MostRestrictedIncl(ListArc, arc$_1$){Keep the most restricted inclusion for a same arc}
    ListCl=MostRestrictedIncl(ListCl, arc$_1$.targetCl){Keep the most restricted inclusion for a same class}
  end for
  nodesIC.remove(0){Remove the class already treated}
end while
```

Analysing a sequence on G_F
If we consider the firing of sequence $s_2 = t_1; t_2; t_4; t_3; t_6; t_5$ in fig. 1.b, corresponding to the loss of a late response, the state class reached is $\mathcal{C}_{F_{23}}$ in G_F (corresponding to node $\mathcal{C}_{21}$ in figures 7.e and 7.f). In Table 1 it appears that $\mathcal{C}_{F_{23}}$ has no corresponding classes for $\alpha > 0.2$ and consequently this means that the possibility of this behaviour is 0.2. For lack of space, the STN $Nt_{i,c}$, delimiting the firing of t_i from class $\mathcal{C}$, for the transitions in the sequence s_2 are given in textual form (see def. 3):
• $Nt_{1,0} : C_{01}(x_0, x_1)$ (arc $a0$);
• $Nt_{2,1} : C_{12}(x1, x2)$ (arc $a29$);
• $Nt_{4,26} : C_{12}(x_1, x_2), C_{14}(x_1, x_4)$ and $C_{24}(x_2, x_4)$ (arc $a28$);
• $Nt_{3,25} : C_{14}(x_1, x_4), C_{13}(x_1, x_3)$ and $C_{43}(x_4, x_3)$ (arc $a27$);
• $Nt_{6,24} : C_{43}(x_4, x_3), C_{46}(x_4, x_6)$ and $C_{36}(x_3, x_6)$ (arc $a26$);
• $Nt_{5,17} : C_{36}(x_3, x_6), C_{35}(x_3, x_5)$ and $C_{65}(x_6, x_5)$ (arc $a20$).
The STN Ns_2 for α-cut .1 and .2 is obtained by the union $Nt_{1,0} \cup Nt_{2,1} \cup Nt_{4,25} \cup Nt_{3,24} \cup Nt_{6,23} \cup Nt_{5,17}$; they have the same structure depicted in figure 9.a. The STN Ns_2 for the core (0-cut) is obtained by the union $Nt_{1,0} \cup Nt_{2,1} \cup Nt_{4,26} \cup Nt_{3,25} \cup Nt_{6,24} \cup Nt_{5,17}$ (respectively, arcs $a0$, $a28$, $a27$, $a26$, $a25$, $a21$ in fig. 7.f) and its structure is also as in figure 9.a. The constraints values

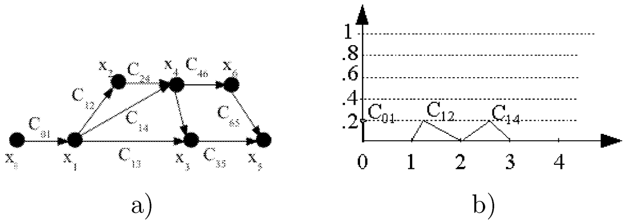


Figure 9: Sequence $s_2 = t_1t_2t_4t_3t_6t_5$: a) STN, b) Fuzzy constraints C_{01}, C_{12} and C_{14}

C_{ij} for all α-cuts are given in Table 2.

arc/α	C_{01}
$a_0/0$:	[0.0 0.0]
$a_0/.1$:	[0.0 0.0]
$a_0/.2$:	[0.0 0.0]

t_1

arc/α	C_{12}
$a_{28}/0$:	[1.0 2.0]
$a_{29}/.1$:	[1.1 1.6]
$a_{29}/.2$:	[1.2 1.2]

t_2

arc/α	C_{12}	C_{14}	C_{24}
$a_{27}/0$:	[1.0 2.0]	[2.0 3.0]	[1.0 2.0]
$a_{28}/.1$:	[1.1 1.6]	[2.3 2.8]	[1.2 1.7]
$a_{28}/.2$:	[1.2 1.2]	[2.6 2.6]	[1.4 1.4]

t_4

arc/α	C_{14}	C_{13}	C_{43}
$a_{26}/0$:	[2.0 3.0]	[3.0 4.0]	[1.0 2.0]
$a_{27}/.1$:	[2.3 2.8]	[3.4 3.9]	[1.1 1.6]
$a_{27}/.2$:	[2.6 2.6]	[3.8 3.8]	[1.2 1.2]

t_3

arc/α	C_{43}	C_{46}	C_{36}
$a_{25}/0$:	[1.0 2.0]	[5.0 5.0]	[3.0 4.0]
$a_{26}/.1$:	[1.1 1.6]	[5.0 5.0]	[3.4 3.9]
$a_{26}/.2$:	[1.2 1.2]	[5.0 5.0]	[3.8 3.8]

t_6

arc/α	C_{36}	C_{35}	C_{65}
$a_{21}/0$:	[3.0 4.0]	[3.4 3.9]	[0.0 1.0]
$a_{20}/.1$:	[3.4 3.9]	[3.4 3.9]	[0.0 0.5]
$a_{20}/.2$:	[3.8 3.8]	[3.8 3.8]	[0.0 0.0]

t_5

Table 2: Constraints values for the STN of sequence s_2 of graphs G^0, $G^{0.1}$ and $G^{0.2}$.

The fuzzy STN $N_F s_2$ (def. 5) is obtained from Table 2 according to property 2. The (triangular) fuzzy constraints such obtained are: $C_{13} = [3\ 3.8\ 4]$, $C_{24} = [1\ 1.4\ 2]$, $C_{35} = [3.4\ 3.8\ 3.9]$, $C_{36} = [3\ 3.8\ 4]$, $C_{43} = [1\ 1.2\ 2]$, $C_{46} = [5\ 5\ 5]$, and $C_{65} = [0\ 0\ 1]$; C_{01}, C_{12} and C_{14} are depicted inf fig. 9.b. We can see that the core of all these constraints is empty and the height is .2. It means that the *possibility* that transition t_6 could be fired is .2 and so the *necessity* (certainty) is zero.

CONCLUSION

This paper has shown that it is possible to use the tool GraphC to analyse a Fuzzy Time Petri net. By decomposing the fuzzy intervals attached to the transitions into a set of α-cuts, generating the state class graphs for each one, and by recomposing the α-cuts, it is possible to derive the fuzzy constraints of the fuzzy simple temporal networks attached to the classes and to the arcs of the fuzzy GraphC.

In place of just answering *yes* or *no* for a property to be verified, it is possible to derive a possibility degree (quantitative view) that the property is not verified. The more important is that the sequences leading to the states expressing the property violation are clearly characterised: the temporal constraints existing among the firing dates are delimited in a complete and minimal way. It will be an important aid for the designer to modify, if necessary, the values of the parameters.

The tool GraphC that generates a graph of classes for a t-time Petri net can be downloaded at `http://graphc.sourceforge.net`. Currently the algorithm 1 is being implemented in order to automatically generate a fuzzy graph of classes. Further work consists in exploring another way to generate the fuzzy graph of classes: its direct generation without using α-cuts.

REFERENCES

[Berthomieu 04] B. Berthomieu, P.O. Ribet, F. Vernadat : The tool TINA: construction of abstract state spaces for Petri nets and time Petri nets, IJPR, Vol.42, N°14, pp.2741-2756, 15th July 2004.

[Cardoso 98] J. Cardoso : Time Fuzzy Petri nets. *Fuzziness in Petri nets*, J. Cardoso and H. Camargo (Ed), Studies in Fuzziness, Physica Verlag, pp 115-145, 1998.

[Cardoso 05] J. Cardoso, S. Cousy, G.Juanole : Extending time Petri nets to fuzzy time Petri nets: definition of the graph of fuzzy state class, 16th IFAC World Congress, July 2005, Prague.

[Dechter 91] R. Dechter, I.Meiri, J. Pearl : Temporal constraint networks, Artificial Intelligence, vol 49, p.61-91, 1991.

[Dubois 88] D. Dubois, H. Prade. *Possibility theory: an approach to computerized processing of uncertainty* Plenum Press, New York, 1988, 263 p.

[Dubois 89] D. Dubois, H. Prade. Processing fuzzy temporal knowledge. *IEEE Trans. on Syst. Man and Cyb.*, 14, 1989, n° 4.

[Mao 05] X. Mao, J. Cardoso, R. Valette : A New Graph of Classes for the Preservation of Quantitative Temporal Constraints. ATVA 2005, Taipei, Taiwan, October 4-7, 2005, LNCS 3707, pp.278-292 Springer-Verlag.

[Merlin 74] P. Merlin. *A Study of the recoverability of Computer Systems.* Phd thesis. University of California, Irvine, 1974.

[Yager 84] R.R. Yager. On different classes of linguistic variables defined via fuzzy subsets. Kibernetes, 13:103-110, 1984.

[Yoneda 98] T. Yoneda, H. Ryuba, CTL Model checking of time Petri nets using geometric regions, *IEICE Trans. inf. & Syst.*, Vol E81-D, No. 3, pp.297-396, 1998.

PETRI NETS SIMULATION

Hybrid Simulation for Critical Scenarios Derivation.

N. SADOU, H. DEMMOU
Laboratoire d'Analyse et d'Architecture des Systèmes
7 avenue du Colonel Roche, F-31077 Toulouse cedex
{sadou, demmou}@laas.fr

Key Words: Hybrid systems, Petri net, simulation, differential equations, Reliability in design.

ABSTRACT

.
This paper illustrates an approach for the safety analysis of systems that are hybrid in nature. It is based on a qualitative analysis of a Petri net model. This method allows deriving feared scenarios by determining the sequences of actions and state changes leading to feared states in which the system fails. The continuous part represented by some differential equations is addressed by local simulation (resolution) of these equations.

1. Introduction

Hybrid dynamic systems (HDSs) are systems with a mix of discrete and continuous components. Depending on the domain of application, HDSs are modeled as dynamic systems with a discrete control [1]. The continuous components are frequently modelled with ordinary differential equations (ODEs). When a discrete event occurs, the system describing a continuous component of an HDS normally changes, and the execution of an HDS changes, or switches discreetly. Hybrid dynamic systems are used, for example, as models of continuous processes controlled by logic controllers or embedded systems. For an overview of HDSs and their applications, see [2].

To address reliability of such system it is necessary to take into account the order of occurrence of events. Classical methods as fault trees [3] are insufficient because they are static. The Dynamic Flowgraph Methodology [4] was introduced in order to take into account the temporal evolution of the system in the fault trees. It is based on a hybrid modelling of the system and an automatic generation of temporal fault trees that is possible with a discretisation of the time and the continuous variables. But as a consequence we have a combinatorial explosion of the states.

This paper presents an extension of the deriving feared scenarios method [5]. It is a qualitative analysis of hybrid systems from the dynamic reliability point of view [6]. It aims to characterise the feared scenarios at the early design stage of the system, to choose the best configuration. The fact that feared scenarios are rare makes the simulation based methods ineffective [7].

A modelling technique associating Petri nets with differential equations [8] has been chosen in order to take into account the hybrid aspect (both continuous and discrete features). The Petri net model describes the operation modes, the failures and the reconfiguration mechanisms. The differential equations represent the evolution of continuous variables of the energetic part of the system. One way to avoid the state space explosion is to directly use the Petri net model to extract the feared scenarios without generating the reachability graph [5]. We use linear logic [9] to get a new representation (based on causality point of view) of the Petri net model, and then extract the scenarios from this new representation. The advantage is that with linear logic we can derive a partial order of transition firings and focus the search on the parts of the model that are involved in a given scenario [5]. This approach is based on the equivalence of reachability in the Petri net and provability of sequents in linear logic [10].

Our modeling approach has the advantage of clearly separating the continuous aspects from the discrete ones. This allows a logical analysis (using linear logic) of the causalities resulting from the state changes, based on the discrete aspect. Thanks to this analysis, and starting from a feared state, it is possible to go back through the chain of causality and to point out only scenarios leading to a feared situation. Each scenario is represented by a partial order between the events necessary to the occurrence of the feared event.

To take into account the continuous dynamic, differential equation solver is used to simulate this dynamic. The resolution of the equation is done only when it is necessary; the deriving feared scenarios algorithm determines which equations will be integrated.

This paper is organized as follows: section 2 introduces the Differential Predicate Transition Petri nets. In section 3 the method for deriving feared scenarios and its algorithm are briefly presented. The section 4 presents the continuous version of the deriving scenario algorithm. An example application is presented in section 5. Finally, section 6 draws a number of conclusions.

2. The modeling approach

2.1 *Differential Predicate Transition Petri Nets (DPT Petri Nets).*

In order to take into account the hybrid aspect (both continuous and discrete features), *differential Predicate Transition Petri Nets* are used. Briefly, a *DPT Petri net* defines an interface between differential equation system and Petri net elements. The Petri net model describes the operation modes, the failures and the reconfiguration mechanisms. The differential equations represent the evolution of continuous variables of the energetic part of the system. Its main features to take into account the continuous part are [8]:

- A set of variables *(x_i)* is associated with each token.
- A differential equation system *(F_i)* is associated with each place *(P_i)*: it defines the dynamics of

the variables associated with the tokens in *Pi*, according to time(θ):

$$F = \begin{bmatrix} F_i(\dot{X}_i, X_i, t) \\ \vdots \\ F_l(\dot{X}_{l,} X_l, t) \end{bmatrix}, \quad i = 1...l$$

- An enabling function *(e*$_i$*)* is associated with each transition *(t*$_i$*)*: it triggers the firing of the enabled transitions according to the value of the xi associated with the tokens of the input places$(X_{input\ _i})$:

  $$e_i(X_{input\ \_i}, \theta) <,=,> 0$$

- A junction function (ji) is associated with each transition (ti): it defines the value xi associated with the tokens of the output places$(X_{output\ _i})$ after transition firing.

  $$X_{output\ \_i}(\theta^+) = j_i(X_{input\ \_i}(\theta^-)) \cdot$$

The advantage of this approach of modeling is that the continuous and the discrete parts are represented by two different formalisms and can be addressed separately.

3. Method for deriving feared scenarios

3.1 *General view of the scenario deriving method*

The method is based on a qualitative analysis initiated from the Petri net model. The aim is to point out the sequence of actions that leads to the feared states and to analyse more precisely what makes the system leave the normal behaviour and reach the feared state. Our method starts by a backward reasoning from the feared state (target state) in order to identify the causal chain of actions leading to that feared state. The backward reasoning is stopped when a nominal state is reached. A forward reasoning follows it in order to obtain all the possible evolutions from this partial nominal state. The bifurcation between the nominal behaviour and the feared one is identified and corresponds to a transition conflict in the Petri net. The analysis of theses bifurcations determines the complete context in which the feared scenario occurs.

When deriving feared scenarios we search for a succession of actions (transition firing), and the necessary context that leads to the marking of places representing the partial feared state. The initial and final marking are partially known. This partial knowledge (partial marking of the Petri net), leads us to progressively enrich the context during the analysis process [14]. This enrichment makes the potentially enabled transitions fireable (it is done by adding token to empty input places of the potentially enabled transition).

Note That the target state to be analysed, can be either a partial feared state or another partial state with a direct or indirect link to the feared state (for example a place that represents the availability of a resource that allows operating with presence of a fault, and avoids the occurrence of the feared event).The 'nominal' places used as stop criteria for a backward reasoning are the places that when marked represents a normal operation state.

3.2 *Principles of the scenario deriving algorithm*

The backward and the forward reasoning are similar and it is why the procedures (algorithm) are the same for both of them [11]. The backward reasoning is done on the reversed Petri net with the target states as initial marking. Forward reasoning evolves in the initial Petri net with the conditioning places as initial marking.

In *Petri net* model, a transition is fireable if it is enabled (its input places are marked). In *DPT Petri net*, the transition must be enabled and the thresholds conditions (enabling function) must be satisfied.

The backward reasoning evolves without taking into account the continuous dynamic of the *PDT Petri net model*. A transition is fireable and fired if all of its input places are marked. If it is potentially enabled (some places are marked and other ones are empty) the empty places are enriched by adding tokens to make it fireable. The determination of the conditioning states is done by reachability analysis between the feared states and the normal sates; transitions are fired until a nominal marking (normal working of the system) is reached.

In the forward reasoning both discrete and continuous evolution guide the scenarios deriving. It is done in the *DPT Petri net*. Tthe conditions related to the continuous thresholds may be satisfied for transition firing. These conditions (enabling functions) are determined according to the functions associated to the input places of the transitions. So it is necessary to solve the differential equations associated to places.

The deriving scenarios algorithm (both backward and forward reasoning) [5] can be considered as a Petri net player. But not classical Petri net player (occurrence graph). It is a player based on linear logic that guides the construction of the partial orders between events [11].

The current version of scenario deriving algorithm is essentially discrete. The continuous part of the system is treated only in the case of linear continuous dynamic. In this case the temporal abstraction is possible. The enabling functions are replaced by temporal thresholds obtained from the resolution of the differential equations associated to places of the *DPT Petri net.* It is done before starting the analysis. This approach is limited to some dynamics and the temporal thresholds (which can be in the form of intervals) are static. These thresholds depend on many factors due to the evolution of the marking of the Petri net and we can't in advance predict this evolution. So it is interesting to determine these thresholds dynamically according to the Petri net marking evolution.

In the next section a new version of the method will be presented. It associates the initial deriving feared scenarios algorithm and a differential equations solver. This association allows us to address complex continuous dynamics and to determine dynamic thresholds.

4. Continuous scenario deriving algorithm

4.1 *Basics*

One of the important issues to tackle when simulating Petri nets is transition conflicts. The way the firing transition is selected in case of conflict has an impact on the evolution of the Petri net. So before presenting the interaction of our algorithm and a differential equations solver we first present a discussion about transition conflict and continuous dynamic. In transitions conflict, considering the thresholds associated to the transitions of a Petri net (i.e. enabling functions), it is difficult to determine a firing order from continuous equations. It is interesting to transform the continuous thresholds (enabling functions) into temporal thresholds. It is possible by simulating (solving) the differential equations associated the different places of the DPT Petri net. Then it is possible to have only one variable (time) which is linear. The linearity of the time variable makes possible to easily define the firing order between transitions, and allows the quantification of the scenarios in terms of time, an important data for the designer.

The resolution of all the differential equations associated to the places of *DPT Petri net*, is time consuming. A solution is to restrict the resolution of the differential equations to some places. The execution of the feared scenario deriving algorithm (that changes the discrete configuration of the system) guides the continuous simulator in the choice of the equations to be solved. So the global simulation is then reduced to a local simulation. The different procedures of the algorithm which can change the discrete configuration of the system are the transition firing, the marking enrichment and the introduction of the initial tokens (initial marking). Following these different procedures the algorithm, recurrently, draws the list of the enabled transitions, according to their firing dates and a firing order is established between them. This order is very important in the reliability analysis of dynamic systems. The firing order which determine the Petri net marking evolution is deduced from the firing dates. The simulation (integration) of the differential equations associated to the input places of the concerned transitions determines these dates and transmits them to algorithm.

The algorithm determines, according to the discrete state of the system, the equations to be integrated and the thresholds to be supervised. The continuous simulator is called and integrates the equations until all the thresholds of the enabled transitions are reached. Then, the simulator transmits the dates to the algorithm, which runs according to the transmitted firing dates. The algorithm uses these dates to determine the evolution of the discrete state and updates the new system of equations to be integrated.

The continuous version of the algorithm is composed by five steps: The first step modifies the structure of the equations system.. The equations associated to the places from which tokens removed are deleted from the system of equations, whereas the equations associated to the places in which a token (or more) is deposited are added to the system of equations. For this step, it is the discrete simulator (scenario deriving algorithm) which is activated, because it updates the marking of the Petri net. We note that the discrete simulator modifies the equations handled by the continuous simulator. Then we enter in new discrete state.

The second step consists on determining the conditions to leave from the current state. The algorithm determines the list of enabled transitions and deduces the threshold events (enabling functions) to supervise. This list is communicated to the continuous simulator.

Then, the third step consists, on executing the junction functions corresponding to the fired transitions. It is necessary to modify the continuous variables values. The integrator is in charge of this step.

The integration of the equations starts and runs until the appearance of all the threshold events (firing dates of the different transitions). This fourth step concerns only the continuous simulator. It is important to note that some thresholds can become true according to the execution of the junction functions j_i, and thus the integration step can be a short-circuit. The transmitted time interval will be [0,0]

Finally when all the thresholds are detected, it is necessary to transmit to the algorithm the different possible firing dates, then the algorithm runs with respect to the transmitted temporal data.

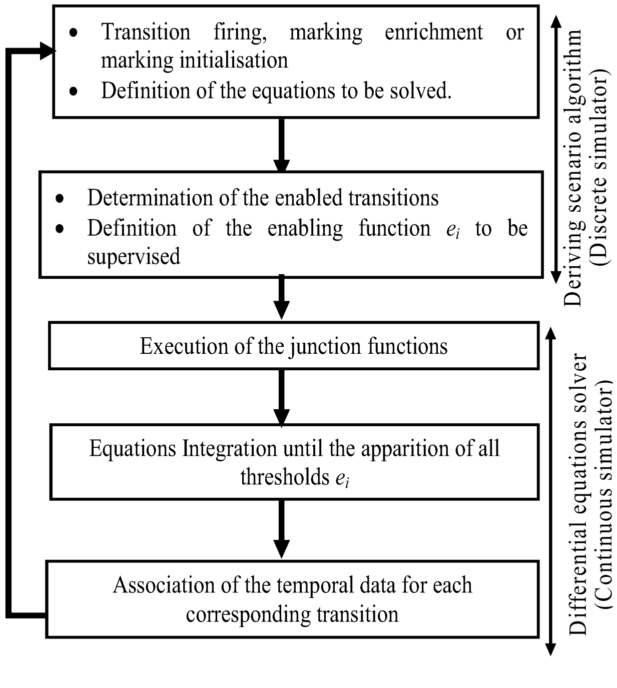

• Transition firing, marking enrichment or marking initialisation
• Definition of the equations to be solved.
• Determination of the enabled transitions
• Definition of the enabling function e_i to be supervised
Execution of the junction functions
Equations Integration until the apparition of all thresholds e_i
Association of the temporal data for each corresponding transition
Deriving scenario algorithm (Discrete simulator)
Differential equations solver (Continuous simulator)

Fig. 1 Coutinuous version of the deriving of feared scenarios algorithm

The principle of the continuous deriving of feared scenario algorithm consists on coupling two simulators. The first one to simulate the Petri net (deriving scenario algorithm). The second to simulate the differential equations systems (integrator). The two simulators evolve alternatively and are synchronized by the events.

In a traditional simulation of *DPT Petri net* [13], the integration of the differential equations stops when the first threshold of the list of the thresholds to be supervised is reached, the discrete simulator is then called in order to fire the corresponding transition. In our approach, the continuous simulator established the firing dates of all transitions of the list provided by the algorithm. The integration stops only when all thresholds are reached. Then firing date are transmitted to establish the firing order of the different transitions.

4.2 *Software implementation*

A prototype of software implementation of the algorithm in the discrete version has been developed using Java [12]. ESA_PetriNet (Extraction Scenarios & Analyzer by PetriNet model) allows deriving from a Petri net model, scenarios that lead to the feared states. For the implementation of the continuous version of the algorithm it is possible to use existing differential equation solver (Matlab, DSolve,…) and implementing interface between ESA_PetriNet and the solver.

5. Case study

5.1 *Presentation*

The case study is based on a volume regulation of a tank (figure2). It consists of a computer, one pump, two electro-valves, one volume sensors, the regulated tank (Tank) and a second tank for draining. The regulated tank is used on demand of a user. This demand is described by a function of time.

The volume of the tank must be kept inside a given interval [V_{min}, V_{max}]. The volume is controlled by the computer, which decides, according to the values given by the volume sensors, to fill (or not) the tank by opening (or not) the electro-valve.

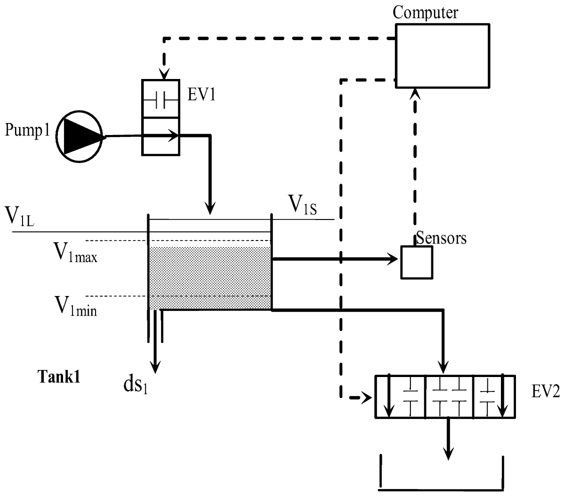


Fig 2 case study

The control law of the computer is such that the electro-valve is closed when the volume of the tank oversteps the upper limit V_{imax}. In the other hand, the computer commands the opening of the electro-valve each time the value of the volume in the controlled tank is lower than the limit V_{imin}. This system must avoid the overflow of the tank. A backup electro-valve is added to the system in order to drain the tanks in case of overflow. When the volume of one tank oversteps the security limit (V_{iL}), the computer commands the opening of the backup electro-valve until the volume becomes lower than V_{imin}. As we focus our study on critical scenarios, and in order to simplify the problem we consider that only the electro-valves can have failures. A typical failure of the electro-valve 1 corresponds to a blocked open state in which the electro-valve does not react to a closing command of the computer. This electro-valve can be repaired after a failure occurrence. When the electro-valve 2 has a failure, it is considered to be definitively out of order.

5.2 *Modelling*

Place *V1_dec* of the net in figure 2 represents the disjunction phase (the pump is closed); place *V1_cr* represents the conjunction phase (the pump is open). Place *EV1_OK* corresponds to a state where the electro-valve 1 works. Transition t_{11} represents the closing command of the electro-valve1 when the volume oversteps *V1max*. Transition t_{12} represents the opening command of the same electro-valve when the volume becomes lower than *V1min*. Transitions *def1* and *rep1* represent the failure of electro-valve 1 (blocked in an open state , and its reparation (*rep1*). When the volume in the tank1 oversteps the high security limit (*V1L*), and the backup electro-valve is available (place *EV2_OK* is marked) then t_{14} becomes fireable and the draining process of tank1 can start via the backup electro-valve by marking place *EV2_oc1*. This phase last the time that it takes for the volume to reach the threshold V1min. Then, the electro-valve2 is released (place *EV2_OK* is newly marked), and a conjunction phase is started again (place *V1_cr* is marked) by firing transition t_{15}. The electro-valve 2 can have a failure (modelled by transition *def2*). In that case, place *EV2_HS* is marked and the electro-valve is set out of order.

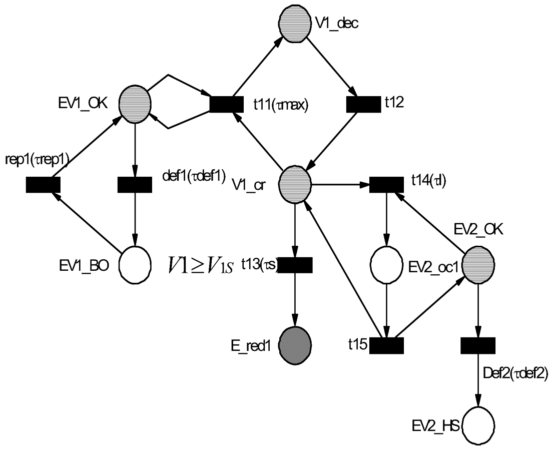


Fig 3. Petri net model of the system

Enabling Function associated to transition:

t_{11} : $v = 100$. t_{14} : $v = 125$.

t_{13} : $v = 150$. t_{12} : $v = 50$.

t_{15} : $v = 50$.

Equation System associated to the place:

$$V1\_cr: \quad \dot{v} = 0{,}02v + 0.2$$

$$V1\_dec: \quad \dot{v} = -0{,}01v + 0.3.$$

5.3 *Scenarios deriving*

The nominal states are the striped places in the Petri net model. We are interested in the overflow of Tank so the target state will be the partial feared state E_red1.

Backward reasoning:

The overflow of tank corresponds to partial marking of the E_red1 place. Thus the backward reasoning starts with one token in this place. By analyzing the inverse Petri net with a token in the *E_red1* place, only the transition t_{13} is fireable. The firing of the transition t_{13} leads to the marking of the *V1_cr* place, which corresponds to a normal place. Thus the conditioning state corresponds to the marking of this place. The backward reasoning stops.

Forward reasoning:

In the backward reasoning step only one conditioning state was found (marking of *V1_cr*); the marking with a token in *V1_cr* will be the initial marking of the forward reasoning.

This step analyses the bifurcations between the normal behaviour and the feared one (firing of the transition t_{13}) that corresponds to the overflow of the tank. When the *V1_cr* place is marked, the transition t_{13} is in conflict with the two transitions t_{11} and t_{14}. So the equation associated to the place *V1_cr* is transmitted to the solver, with the thresholds to supervise (e_{11}, e_{14} and e_{13}). These thresholds correspond to the enabling function of the transitions $t_{13},$ t_{11} and t_{14}. The integration of the equation associated to the place *V1_cr* starts. The firing dates determined by the continuous simulator (we use matlab differential equations solver) are T_{imin}= 30 *sec* for t_{11}, T_{iL} = 40 *sec* for t_{14} and T_{is} = 45 *sec* for t_{13}. So $T_{imin},< T_{iL} < T_{is..}$ These firing date are transmitted to the discrete simulator (ESA_PetriNet).

By analysing (using ESA_PetriNet) the conflict between the transitions $t_{11},$ t_{14} and t_{13} and according to the temporal transmitted data, the transition t_{13} cannot be fired before the transitions t_{11} and t_{14} because its temporal threshold is higher than those of the transitions t_{11} and t_{14}.

The occurrence of the feared event (firing of the transition t_{13}) is necessarily the consequence of the non-firing of t_{11} and t_{14}. These two transitions (t_{11} and t_{14}) are potentially enabled. To determine why these two transitions are not fired, it is necessary to:

- Enrich the marking of their input places; one token is added to the place *EV1_OK* and another one to *EV2_ok*.
- And forbid the firing of the transitions t_{11} and t_{14}.

Now, two transitions are fireable; *def1* and *def2*. After their firings, only the transition t_{13} can be fired. Its firing leads the system to the feared state that corresponds to the marking of the place *E_red1*.

The feared scenario could now be characterized. It is composed of the following events:

Scenario : failure of electro-valve 1 (transition *def1*) failure of the backup electro-valve (transition *def2*), followed by the overflow of tank (transition t_{13})

The figure4 is a precedence graph (partial order graph) that represents the feared scenario that corresponds to the overflow of the tank1. It shows that the failure of the *EV1* (transition def1) and the EV2 one (transition def2) precede the overflow of tank1event transition T13). The events E1 (respectively E2) represents the enrichment of marking of place EV1_0 (respectively EV3_OK) while I1 represents the initial event corresponding to the presence of a token in the V1_cr place produced in the backward reasoning step. The discontinuous arrows represent the orders between the events of the scenario.

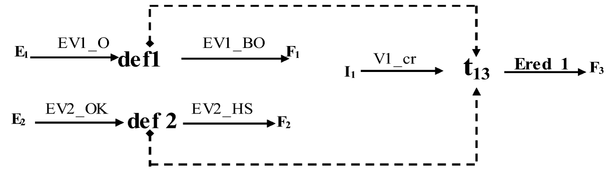


Fig. 4. Feared scenario

6. CONCLUSION

The approach that we have presented in this paper is an extension of the deriving feared scenario method. This extension allows addressing hybrid systems reliability.

The *DPT Petri net* modelling approach in which the continuous and the discrete parts are represented by two different formalisms has the advantage to clearly separate these two aspects. It allows a logical analysis (using linear logic) of the causalities resulting from the state changes. Thanks to this analysis, and starting from a feared state, it is possible to go back through the chain of causality and to point out all the possible scenarios leading to a feared state. Each scenario is given by a partial order between the events necessary to the occurrence of the feared event. The order between events is related to causal relation in the Petri net and imposed by the continuous dynamic of the system.

The causal relations are determined by combining the initial deriving feared scenarios algorithm (discrete simulator) and the differential equations solver (continuous simulator). These two simulators evolves alternatively, the discrete simulator determines the state changes according to the timed data transmitted by the continuous simulator.

References

[1] A. Gollu and P. Varaiya. Hybrid dynamical systems.In Proc. 28th IEEE Conf. Decision Contr., pages 2708-2712, 1989. Tampa, Florida.

[2] R. Alur, T. Henzinger, G. Lafferriere, and G.Pappas. Discrete abstractions of hybrid systems. In Proceedings of IEEE, volume 88, pages 971-984, 2000.

[3] Lee, W. S.; Grosh, D. L.; Tillman, F. A., Lie, C. H., "Fault tree analysis, methods, and applications – A

review", IEEE Transactions on Reliability, August 1, 1985; ISSN 0018-9529; r-34, page 194-203.

[4] Chris J. GARRET, Sergio B. Guarro, George E. APOSTOLAKIS, "The Dynamic Flowgraph Methodology for Assessing the Dependability of Embedded Software Systems", IEEE Transactions On Systems, Man, and Cybernetics, Vol. 25, No. 5, May 1995.

[5] Sadou N, H Demmou, J. C Pascal , R Valette. Object oriented approach for deriving feared scenarios in hybrid system. 2005 European Simulation and Modelling Conference, Porto (Portugal), 24-26 October 2005, pp.572-578.

[6] F. Dufour, Y. Dutuit, "Dynamic Reliability: A new model", 13-ESREL2002 European Conference, Lyon - France - 18 au 21 Mars 2002.

[7] P.E. Labeau: "A Survey on Monte Carlo Estimation of Small Failure Risks in Dynamic Reliability". In International Journal of Electronics and Communications, Vol. 52, pp. 205-211, 1998.

[8] R. Champagnat, P. Esteban, H. Pingaud, R. Valette, " Modelling and simulation of a hybrid system through Pr/Tr PN DAE model ", ADPM'98 3rd International Conference on Automation of Mixed Processes, 19-20 March 1998, Reims, France p. 131-137.

[9] J.Y Girard, " Linear Logic ", Theoretical Computer Science, 50, 1987, p.1-102.

[10] B. Pradin-Chézalviel, R. Valette, L.A. Künzle: "Scenario duration characterization of t-timed Petri nets using linear logic", IEEE PNPM'99, 8th International Workshop on Petri Nets and Performance Models, Zaragoza, Spain, September 6-10, 1999, p.208-217.

[11] H. Demmou, S. Khalfaoui, N. Rivière, E. Guilhem, « A method forderiving critical scenarios from mechatronic systems », Journal Européen des Systèmes Automatisés, volume 36 – n°7/2002, pages 987 à 999.

[12] Malika Medjoudj 'Contribution à l'analyse des systèmes pilotés par calculateurs : Extraction de scénarios redoutés et vérification de contraintes temporelles', thèse de Doctorat, Université Paul Sabatier, Toulouse, 2006.

[13] Ronan Champagnat 'Supervision des systèmes discontinus : définition d'un modèle hybride et pilotage en temps-réel', thèse de Doctorat, Université Paul Sabatier, Toulouse, 1998.

[14] N. Sadou, H. Demmou, J.C. Pascal, R. Valette. Fiabilité dynamique des systèmes hybrides : Approche Basée Scénarios. Conférence internationale francophone d'automatique. Bordeaux (France) 30 Mai-1 juin 2006..

EFFICIENT ENABLING TEST IN SIMULATION OF SWN

Lorenzo Capra
Università degli Studi di Milano
Dipartimento di Informatica e Comunicazione
via Comelico 39/41, I-20135 Milano, Italy
E-mail:capra@dico.unimi.it

Massimiliano De Pierro
Università degli Studi di Torino
Dipartimento di Informatica
via Pessinetto 12 , 10149 Torino, Italy
E-mail:depierro@di.unito.it

ABSTRACT

Recently, in the area of modelling and analysis with Stochastic Well-formed Net it has been introduced a framework that shows itself promising for efficient structural analysis. This paper, exploiting such framework, considers a very common task among SWN analysis methods: The computation of the transitions enabled in a given marking. This task does not affect the reachability graph construction only, but also methods based on model-checking, and especially the discrete-event simulation, which is an interesting alternative when exact stochastic Markovian solutions of SWN are hard to obtain due to the state-space explosion problem or when more general probabilistic distributions are used in the model description.

INTRODUCTION

Stochastic Well-formed Coloured Net (SWN) (Chiola et al., 1993) is a Petri Net (PN) formalism for the modelling of discrete-event systems such as computer networks, protocols, distributed algorithms, workflow and so on. SWN evolved from PN and belongs to the category of High-level Petri Net (Jensen and Rozenberg, 1991). Modern computer-systems show marked symmetries in their design and behaviour. SWN catches such symmetries in a structured and constrained syntax allowing modellers to manage the complexity of the design and providing a more "natural" formal description than ordinary PN. Although the formalism's expressiveness increases in quality, the analysis techniques, to be efficient, should exploit such higher-level model description. In this effort several analysis algorithms have been efficiently developed exploiting SWN's symmetries (Chiola et al., 1997). Most of them are however based on good representations of the state-space. Only recently, (Capra et al., 2005) proposed a promising framework that allows to implement algorithms needing some sort of SWN structural analysis. It is remarkable however that also state-space based methods can take advantages of such framework because often they require some consideration about the SWN net structure.

This paper focuses on the problem of computing the enabled instances of a SWN's coloured transition in a given marking. This task is relevant in many SWN analysis techniques because it is strictly bound to the behavioural analysis of a model. Here the attention is on the simulation of a system with a SWN model. (Chiola and Gaeta, 1995; Gaeta, 1996) point out how efficiency in the *enabling test* is relevant in order to reduce the temporal overhead in performance analysis carried out by means of discrete-event simulation using SWN.

This paper shows an algorithm based on the SWN framework to improve in time the enabling test in SWN. The approach is denoted as symbolic compared to such methods that utilise unfolding into PN: it essentially consists in manipulating the model's arc functions according to rewriting rules based on algebraic properties, so to find formal expressions for local enabling functions whose syntax and semantics is that of the SWN formalism.

The paper is so organised into its main sections: first it recalls in a pragmatic way the SWN formalism. Next, the enabling test problem in SWN is discussed, and the state-of-the-art with regard to several techniques that can be utilised to deal with it are provided. Next section recalls the theoretical framework the symbolic approach here proposed is based on. The last two sections are the paper's contribute: They describe the proposed algorithm with the aim to solve in an efficient way the enabling test task in SWN, discussing the symbolic enabling test with the symbolic marking notion, and provide an application to an example taken from the literature. An outline of the presented work and a discussion on possible evolutions end the paper.

THE SWN FORMALISM

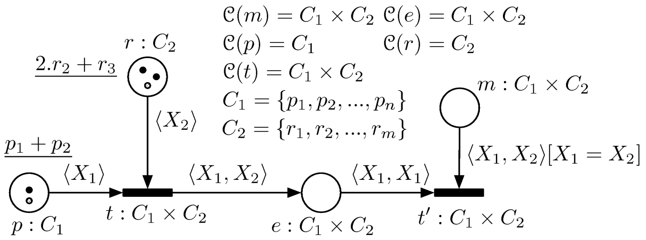
Formally a SWN model is a tuple

$$(P, T, \{C_1, ..., C_n\}, \mathcal{C}, W^-, W^+, H, \mathbf{m}_0) \qquad (1)$$

The paper does not illustrate all the features of the formalism - the interested reader may found a complete description in the original work (Chiola et al., 1993) - but just introduces Definition (1) informally using the model depicted in Fig. 1 as relief. In Definition (1) P is the finite set of net's *places*. With respect to PN, places are not marked by indistinguishable tokens, but may contain tokens of different identities, called colours and ranging on sets called colour domains. A marking $\mathbf{m}$ maps each place to a multiset of colours belonging to the place colour domain. $\mathbf{m}_0$ defines the initial marking of the net. In the example place r initially contains the multiset of colours $\{r_2, r_2, r_3\}$ which denote two resources of kind r_2 and one of kind r_3, while place p two processes, p_1 and p_2. Multisets of colours are denoted by formal sum, so it is written $\mathbf{m}_0(r) = 2.r_2 + 1.r_3$ and $\mathbf{m}_0(p) = 1.p_2 + 1.p_1$. Cartesian products of basic classes of colours, C_1 and C_2 in the example, define the colour domain of the places. Actually in SWN any colour class C_i may be partitioned into sub-classes denoted $C_i = \bigcup_j C_{i,j}$. In Definition (1) T is the finite set of the net's *transitions*. Transitions are coloured too, and their colour domains are implicitly derived from the functions labelling the surrounding arcs. A coloured transition actually folds together many elementary ones: in SWN by *instance* of colour c of a coloured transition t it is denoted the elementary transition t_c corresponding to a PN transition. Transition t in the example of Fig. 1 represents many elementary instances, each one denoted by a pair (p_i, r_j). Function $\mathcal{C}$ assigns to each $s \in P \cup T$ a colour domain C_i. Notation $Bag(C_i)$ denotes the set of all possible multisets that can be built on elements in C_i. The *support* operator maps a multiset into the corresponding set containing the same colours but with multiplicity 1 only, for instance $\underline{2.r_2 + 1.r_3} = 1.r_2 + 1.r_3$.

W^-, W^+ and H are respectively the input, output and inhibitor arc functions, and are mappings assigning to each pair $(t,p) \in T \times P$ a function $F(t,p) : \mathcal{C}(t) \to Bag(\mathcal{C}(p))$, in turn $F(t,p)$ assigns to each instance of t a multiset on the colour domain of place p. SWN formalism imposes several constraints on the arc function syntax. A colour class-*i* function f_j is a mapping $\mathcal{C}(t) \to Bag(C_i)$ and can be an integer linear combination of elementary functions denoted by: X_k, S, $S_{C_{i,j}}$, $S - X_k$. A *function tuple* $\langle f_1, \ldots f_n \rangle$ is a tuple of class functions. In the example $\langle X_1, X_2 \rangle$ is a function tuple where the first component is a class-*1* function and the second one is a class-*2* function. The angular brackets in tuples denote a Cartesian product so for instance the codomain of the function-tuple $\langle X_1, X_2 \rangle$ is $Bag(C_1) \times Bag(C_2)$. In SWN a guard $[guard]$ is a function $[guard] : \mathcal{C}(t) \to \mathcal{C}(t)$ useful to filter out from the domain of a transition t those instances which do not satisfy a given Boolean predicate. In the example, function $W^-(t', m)$ is guarded by $[X_1 = X_2]$ by means of the composition $W^-(t', m) \circ [X_1 = X_2]$. The Boolean predicate is built upon a limited set of basic predicates: $X_i = X_j$, $d(X_i) = d(X_j)$, $X_i \neq X_j$, $d(X_i) \neq d(X_j)$ - where $d(X_i)$ denotes the codomain of X_i. Finally, $W^-(t,p)$,$W^+(t,p)$ and $H(t,p)$ are integer linear combinations of guarded function tuples. Guards may be also associated to a transition t to restrict its set of admissible colour instances. The semantics of the elementary symbols is: X_k projects the arguments on its k-*th* components, when $d(X_k)$ is circularly ordered a d^{th}-successor operator $!^d$ may be applied to X_k; S is a constant mapping each element of its domain to the multiset C_i; $S_{i,j}$ is a constant mapping to the multiset $C_{i,j}$, assumed class C_i partitioned in sub-classes; $S - X_k$ is the complement of the k-th component of the argument. Referring to the example of Fig. 1, the application of $\langle X_1, X_2 \rangle$ to the colour instance (p_i, r_j) of t results in $\langle X_1, X_2 \rangle (p_i, r_j) = X_1(p_i, r_j) \times X_2(p_i, r_j) = (p_i, r_j)$.

A reachability-graph representing the model behaviour in terms of states and transitions between states is built starting from the initial marking $\mathbf{m}_0$ and firing the *enabled transition set*. In stochastic extensions of the formalism such graph is associated to a Continuos Time Markov Chain (CTMC) which solution leads to compute the performance indices of interest. When exact solution of the Markovian process is not feasible or the probabilistic distributions involved into the model do not lead to an underlying Markovian process then discrete-event simulation can be utilised. The SWN arc functions syntax allows system symmetries to be encoded into the SWN model, in this way efficient methods exploiting SWN's symmetries can be applied to build a compact Symbolic Reachability Graph (SRG) and a corresponding stochastic process (*lumped* CTMC), or to perform symbolic discrete-event simulation runs (Chiola et al., 1997).



$\mathcal{C}(m) = C_1 \times C_2$ $\mathcal{C}(e) = C_1 \times C_2$
$\mathcal{C}(p) = C_1$
$\mathcal{C}(r) = C_2$
$\mathcal{C}(t) = C_1 \times C_2$
$C_1 = \{p_1, p_2, ..., p_n\}$
$C_2 = \{r_1, r_2, ..., r_m\}$
$r : C_2$
$\underline{2.r_2 + r_3}$
$\langle X_2 \rangle$
$\underline{p_1 + p_2}$
$\langle X_1 \rangle$
$\langle X_1, X_2 \rangle$
$\langle X_1, X_1 \rangle$
$m : C_1 \times C_2$
$\langle X_1, X_2 \rangle [X_1 = X_2]$
$p : C_1$
$t : C_1 \times C_2$
$e : C_1 \times C_2$
$t' : C_1 \times C_2$

Figure 1: Resource Acquisition with SWN

THE ENABLING TEST

The determination of the enabled instances of a transition in a given marking is a common task for those analysis techniques needing to explore the state-graph of a PN model. When the formalism used to build the models is coloured as in case of SWN, this task might be efficiently performed exploiting the structured syntax of the paradigm. The problem may be stated in terms of looking for a function $EN(\mathbf{m}, t)$ which for a given marking $\mathbf{m}$ and transition t gives the enabled instances of t. Implementing this function in an efficient way is a dominance factor in making the algorithms based on state exploration faster (Gaeta, 1996).

This paper introduces a method to compute the enabled instances of coloured transitions without unfolding the coloured net but manipulating the arc functions in a symbolical way. That is achieved exploiting the results presented in (Capra et al., 2005).

Enabling Test in SWN

In SWN the straightforward way to test the enabling of transitions in a given marking is the SWN unfolding, that is the SWN is implicitly translated into a PN model and the test is done on this latter. The condition to test to state if transition instance (t, c) is enabled in marking $\mathbf{m}$ is:

$$W^-(t,p)(c) \leq \mathbf{m}(p) < H^-(t,p)(c), \ \forall p \qquad (2)$$

where the comparison operators are on multisets. Then, $EN(\mathbf{m}, t)$ is obtained by evaluating (2) for each colour c of t. This kind of method is numeric and does not exploit the symmetries encoded in coloured functions $W^-(t,p)$, $W^+(t,p)$ and $H(t,p)$. The literature presented several works which exploit SWN's features in order to enhance the calculus of $EN(\mathbf{m}, t)$. The optimisation proposed in (Gaeta, 1996) was integrated in the *GreatSPN* tool and improves the efficiency of the discrete-event simulation engine for SWN models. This optimisation implements a series of heuristics that may be applied when SWN arc functions match particular patterns only. For instance function tuples can not include the complement function, and can not be associated to guards. Transition guards are only partially treated. The proposed approach thus exploits SWN features for a restricted set of cases, characterised by colour annotations having a simpler structure than this paper foresees. A second work on this topic is (Jean-Michel Ilie and Omar Rojas, 1993) and it is more related to the contents of this paper presenting some similarities. It provides a basic calculus for SWN enabling test, however this is not symbolic in all steps, requiring to consider colour identities. For sake of completeness there exists a different kind of methods (Evangelista and Pradat-Peyre, 2004) which optimise the $EN(\mathbf{m})$ computation exploiting structural information of the net such as *causality* and *conflicts* in order to efficiently update the set of enabled transitions after any transition firing so to limit the transitions to consider at each step of the generation of the state space. However also these methods finally require to test condition (2).

This paper presents an enhancement of (Jean-Michel Ilie and Omar Rojas, 1993) work. Exploiting a calculus recently presented in (Capra et al., 2005) it is shown how the technique may be generalised to comprise all SWN features and this may be done symbolically, i.e. without unfolding of places and transitions. The hypothesis is that W^-,W^+ and H may be manipulated by means of operators working on function tuples in order to obtain a final functional expressions which returns the enabled coloured instances of a transition t given a marking $\mathbf{m}$.

THE SYMBOLIC MANIPULATION FRAMEWORK

(Capra et al., 2005) introduces a framework in the effort to provide an engine of calculus upon which to implement SWN analysis algorithms that require some structural analysis. The framework is characterised by a high-level language to express SWN structural relations, such as conflict and causal connection, and several operators on it. The language syntax is a simple extension of SWN arc function syntax. Thus $W^-(t,p)$, $W^+(t,p)$, $H(t,p)$ belong to the language. The operators are the ones needed by most of the algorithms based on structural check of SWN.

To compute $EN(\mathbf{m})$ two main operators are utilised: the transpose ^{t}, and the difference $\ominus$. Besides them, which are top-level operators on the language, there are several ones defined on each semantic object: The smallest ones, the language's tokens, are the SWN elementary functions X_k, S_i, $S - X_k$, $S_{i,j}$ and $!^h$.

The Language of the Symbolic Expressions

The language $\mathcal{L}$ is a set of expressions E_i closed with regard to a given collection $\mathcal{O}$ of operators. If $*$ is a binary operator in $\mathcal{O}$, the skill of the symbolic calculus is to compute for each $E_i, E_j \in \mathcal{L}$ an expression $E_k \in \mathcal{L}$ such that: $E_k = E_i * E_j$.

Syntactically, an expression $E \in \mathcal{L}$ is a formal sum of terms $[f_i] \circ T_i \circ [g_i]$ where $[g_i]$ and $[f_i]$ are SWN's guards and T_i is a SWN's function tuple with some more extension. An example of expression in $\mathcal{L}$ is the guarded tuple $\langle 2.X_1 + (S - X_2), !^3 X_1\rangle[X_1 \neq X_2]$ - the composition symbol $\circ$ is usually omitted. With respect to the original SWN syntax, guards may be left-applied to function-tuples, in such a case they are called *filters*. In order to satisfy the closure requirement under set $\mathcal{O}$, an extension has been necessary: class-functions are no more linear combinations of elementary functions only, but may be linear combinations of their *intersections*, so $\langle X_1 \cap !^2 X_2 + (S - X_2), X_1\rangle$ belongs to $\mathcal{L}$ too. The extensions above make the SWN syntax richer from a descriptive point of view.

It is observable that different expressions $E_i, E_j \in \mathcal{L}$ may denote the same function. The subset of $\mathcal{L}$ composed by $E_i \in \mathcal{L}$ such that class functions are intersections of elementary symbols, is regarded as the kernel set and is denoted as $\mathcal{K}$. The following property states the equivalence between $\mathcal{K}$ and $\mathcal{L}$.

Prop. 1 *For each $E \in \mathcal{L}$ there exists $E' \in \mathcal{K}$ such that $E' = E$, where $=$ denotes the equality between functions.*

The framework introduces the rewriting rules translating E to E'. A useful property of tuples forming an expression E' is that they map on sets, thereby a sum of tuples may be easily translated to a pairwise disjoint sum. The enabling test algorithm presented next requires this assumption to operate.

The Language Operators

Def. 1 *Transpose* $(\cdot)^t$ - In SWN a function labelling an arc (t,p) is a mapping $\mathcal{C}(t) \rightarrow Bag(\mathcal{C}(p))$. Let us focus on $W^-(t,p)$ and consider the following operation on it: the operator ^{t} applied to $W^-(t,p)$ results in a function $W^-(t,p)^t : \mathcal{C}(p) \rightarrow Bag(\mathcal{C}(t))$ such that $W^-(t,p)^t(c')(c) = W^-(t,p)(c)(c')\ \forall c \in \mathcal{C}(t), \forall c' \in \mathcal{C}(p)$ (assuming m to be a multiset, $m(a)$ denotes the multiplicity of element a in m). The operator ^{t} is called the transpose. If one knows $W^-(t,p)^t$ then, given a p's colour instance c, $W^-(t,p)^t(c)$ maps to the t's instances which need c on place p in order for being enabled, along with its quantity. For example a function $W^-(t,p)^t = \langle S - X_1, 2X_1\rangle$ from $\mathcal{C}(p)$ to $Bag(\mathcal{C}(t))$ when evaluated in $r_1 \in C_1$ maps to multiset $\langle C_1 - r_1, 2.r_1\rangle = 2.\langle C_1 - r_1, r_1\rangle$. It means that t instances in $\langle C_1 - r_1, r_1\rangle$ require two tokens r_1 on place p to be enabled.

Def. 2 *Difference* $\ominus$ - Let $a = \sum_{\forall c \in C} \lambda_i.c$ and $b = \sum_{\forall c \in C} \gamma_i.c$ be multisets on C; then $a \ominus b$ is so defined: $a \ominus b = \sum_{\forall c \in C} sup[0, \lambda_i - \gamma_i].c$. Let F and F' be functions from C to $Bag(D)$ and $c \in C$; then $(F \ominus F')(c) = F(c) \ominus F'(c)$.

Def. 3 *Intersection* $\cap$ - If functions are on sets then $(F \cap F')(c) = F(c) \cap F'(c)$. F, F' are said to be disjoint if $F \cap F'$ is equal to the null function.

Each framework's operator is bound to some rules which rewrite expressions of $\mathcal{L}$ to obtain the result. These rewritings are justified by several algebraic properties the expressions in $\mathcal{L}$ satisfy and whose theoretical foundation is given in (Capra et al., 2005). Hereafter symbol $\rightarrow$ is utilised to denote the application of a framework's rewriting rule. Next, there are two properties which justify some of the rewritings the algorithm which compute the enabled instances utilises.

Prop. 2 *Let f and g be functions mapping to sets and $\lambda, \gamma \in \mathbb{N}$, then it holds:*

$$\lambda f + \gamma g = \lambda(f \ominus g) + (\lambda + \gamma)(f \cap g) + \gamma(g \ominus f)$$

and the addends $(f \ominus g)$,$(f \cap g)$,$(g \ominus f)$ are disjoint.

Property 2 states that any sum of functions mapping on sets, can be rewritten into an equivalent sum of pairwise disjoint terms. As an example, utilising Prop. 2 to the functions appearing in tuple $\langle 2S + X_1, X_2\rangle$ this can be rewritten as $\langle 2(S - X_1) + (X_1 - S) + 3(S \cap X_1), X_2\rangle$, then the tuple becomes $\langle 2(S - X_1) + 3X_1, X_2\rangle$. Expansion now gives $2\langle S - X_1, X_2\rangle + 3\langle X_1, X_2\rangle$, that belongs to $\mathcal{K}$.

For functions mapping on sets it holds: $f \ominus g = f \cap {}^c g$, where the complement ^{c} of a function is defined as $({}^c f)(a) = {}^c(f(a))$ and its linear extension is defined as $({}^c f)(A) = \sum_{a\in A}({}^c f)(a))$.

Prop. 3 *Let* $T = [out]T'[in] \in \mathcal{K}$, $T' = \langle f_1, f_2, \ldots, f_n\rangle$ *Then* ${}^cT$ *is equivalent to*

$$[\neg out]\langle \otimes_{j=1}^{n} S\rangle + [out]\langle \otimes_{j=1}^{n} S\rangle[\neg in] + \sum_{i=1}^{n} G_i,$$
$$G_i = [out]\langle \otimes_{j=1}^{i-1} f_j, {}^c f_i, \otimes_{j=i+1}^{n} S\rangle[in]$$

Property 2 and Prop. 3 allow to translate any expression $E \in \mathcal{K}$ into an equivalent $E' \in \mathcal{K}$ sum of pairwise disjoint terms, moreover since empty function ϕ representation in $\mathcal{L}$ is unique it is possible to state the equivalence between generic functions: $T_1 \equiv T_2 \Leftrightarrow T_1 \ominus T_2 = T_2 \ominus T_1 = \phi$.

COMPUTING $EN(\mathbf{m}, t)$ IN SWN

One who looks at Condition (2) may observe that the computation of $EN(\mathbf{m}, t)$ using such definition is highly inefficient because it is focused on transition instances: given any place $p \in P$, Condition (2) requires to consider any colour c of t and to check if the marking satisfies the multiset inequality obtained applying functions $W^-(t,p)$ and $H(t,p)$ to c. This way to proceed implicitly considers marking $\mathbf{m}$ many times in the calculation, one for each colour instance of t - the calculus is redundant. It is more useful to have a function that applied to a *colour place* of p gives the instances of the transition t the colour may enable. This kind of function exists and it is the transpose of the arc function.

The computation of $EN(\mathbf{m}, t)$ is illustrated as a two steps procedure: The computation of a functional expression of *locally* enabled instances denoted with $\mathcal{LE}(t,p)$ that states the potentially enabled instances of t due to place p - the expression is independent form marking considerations; The use of $\mathcal{LE}(t,p)$ in a given marking $\mathbf{m}$ to compute $E(\mathbf{m}, t)$. Given a transition t and a place p, the function $\mathcal{LE}(t,p) : \mathcal{C}(p) \rightarrow 2^{\mathcal{C}(t)}$ represents the locally enabled set function and it is such that when applied to a given colour in place p it gives the set of colour instances of t that are enabled as concerns p and colour c. Following from Definition (2) it may be verified that the set of colour instances of transition t enabled in a given marking is given by intersecting the sets resulting from $EN(\mathbf{m}, t)$ when computed for each $p \in P$.

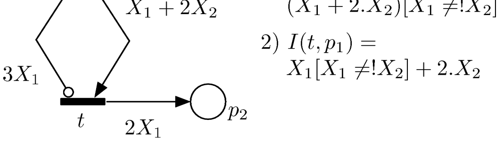


Figure 2: The Running Example: a SWN Model

Figure 2 shows the running example utilised through the next sections to illustrate the application of the method.

Locally Enabled Set Computation

In order to compute function $\mathcal{LE}(t,p)$ it is necessary to consider the coloured tokens occurring on p at $\mathbf{m}$ and evaluate which instances of t they enable. Such information are contained in the transpose of functions $W^-(t,p)$ and $H(t,p)$, that map from colour domain of place p to multisets on colour domain of transition t.

Consider $W^-(t,p)^t$. Its application $W^-(t,p)^t(c)$, $c \in \mathcal{C}(p)$, results in a multiset $a = \alpha_1.c'_1 + \ldots + \alpha_n.c'_n$ on $\mathcal{C}(t)$, $n = |\mathcal{C}(t)|$. The coefficients $\alpha_i \in \mathbb{N}$ provide the information about the requirements imposed by the input function $W^-(t,p)$ on every instance c'_i of t, respective to colour c: precisely, there must be at least α_i occurrences of c in p for c'_i instance of t to be enabled in the given marking. Applying the same reasoning to inhibition function $H(t,p)^t$ it is possible to write an enabling table, which enumerates for each $c'_i \in \mathcal{C}(t)$, and for each $c \in \mathcal{C}(p)$, an interval of positive integer values representing the number of occurrences of colour c in p that ensures the local enabling of instance (c'_i) of t. The first step consists in finding compact representations for local enabling tables using transposition and difference operators on language $\mathcal{L}$.

Transposing $W^-(t,p)$ and $H(t,p)$
The symbolic framework provides efficient algorithms to compute the transposes of $W^-(t,p)$ and $H(t,p)$. It is assumed that both $W^-(t,p)^t$ and $H(t,p)^t$ are given in the following form:

$$\begin{array}{ll} W^-(t,p)^t & = \lambda_1 I_1 + \ldots + \lambda_m I_m \\ H(t,p)^t & = \gamma_1 H_1 + \ldots + \gamma_n H_n \end{array}$$

where $\lambda_i, \gamma_i \in \mathbb{N}$ and I_i, H_i are guarded and filtered function tuples belonging to $\mathcal{K}$: formally $[f_i]T_i[g_i]$. The algorithm here introduced is based on rewriting both $W^-(t,p)^t$ and $H(t,p)^t$ in sums of disjoint terms. By means of Prop. 2, which is based on difference operator, it is possible to operate the symbolic framework to obtain such expressions.

In the net of the example of Fig. 2, the transposes of $W^-(t,p_1)$ and $H(t,p_1)$, that are functions from C_1 to $C_1 \times C_1$, are respectively:

$$\begin{array}{l} \langle X_1\rangle^t + 2.\langle X_2\rangle^t \rightarrow \langle X_1, S\rangle + 2.\langle S, X_1\rangle \\ 3.\langle X_1\rangle^t \rightarrow 3.\langle X_1, S\rangle \end{array}$$

Applying the disjoining steps based on Prop. 2 to $W^-(t,p_1)^t$ and using the rewriting rules of operator $\cap$, it results in the expression for $W^-(t,p_1)^t$:

$$\langle X_1, S\rangle + 2.\langle S, X_1\rangle \rightarrow 3.(\langle X_1, S\rangle \cap \langle S, X_1\rangle) + 1.(\langle X_1, S\rangle \cap \langle S, S - X_1\rangle) + 2.(\langle S, X_1\rangle \cap \langle S - X_1, S\rangle) \rightarrow 3.\langle X_1, X_1\rangle + 1.\langle X_1, S - X_1\rangle + 2.\langle S - X_1, X_1\rangle$$

Disjoining the terms of both transposes allows the algorithm to test the enabling with regard colour multiplicity in p.

$$\begin{array}{ll} W^-(t,p_1)^t & = 3.\langle X_1, X_1\rangle + 1.\langle X_1, S - X_1\rangle + 2.\langle S - X_1, X_1\rangle \\ H(t,p_1)^t & = 3.\langle X_1, S\rangle \end{array}$$

The Tabular Symbolic Form
In the last two computed expressions, functional term $\langle X_1, S\rangle$ of $H(t,p_1)^t$, when applied to a same

colour c, refers to instances that are contained either in $\langle X_1, X_1\rangle$ or in $\langle X_1, S - X_1\rangle$ of $W^-(t, p_1)^t$. A more convenient form for $H(t,p)^t$ is given by $3.\langle X_1, X_1\rangle + 3.\langle X_1, S - X_1\rangle + \infty.\langle S - X_1, X_1\rangle$ which can be easily verified to be equivalent to $\langle X_1, S\rangle$ and has got the same functional terms as $W^-(t, p_1)^t$. Expressing $H(t, p_1)^t$ in this way allows a single expression to be written which states the enabling of t due to both inhibition and input conditions.

$$\mathcal{LE}(t, p_1) = \underbrace{\langle X_1, X_1\rangle}_{[3,3)} + \underbrace{\langle X_1, S{-}X_1\rangle}_{[1,3)} + \underbrace{\langle S{-}X_1, X_1\rangle}_{[2,\infty)}$$

$\mathcal{LE}(t, p_1)$ denotes the locally and potentially enabled instances of t: *locally* means that the instances are only those due to place p_1, while *potentially* means that they are a super-set of the ones actually enabled and other considerations have to be done in order to compute $EN(\mathbf{m}, t)$. For instance, taking into account bounds, the second term of the expression says that tokens of colour c in place p may enable the t's instances $\langle c, C_1 - c\rangle$ iff they are in a number of at least 1 and at most 2.

Rewriting of $H(t, p_1)^t$ to meet the tabular representation can be automatically done using the framework's difference operator. The following steps show the application of the algorithm to the example:

$3.\langle X_1, S\rangle \rightarrow 3.(\langle X_1, S\rangle \cap \langle X_1, X_1\rangle) + 3.(\langle X_1, S\rangle \cap (\langle S - X_1, S\rangle + \langle X_1, S - X_1\rangle)) = 3.\langle X_1, X_1\rangle + 3.\langle X_1, S - X_1\rangle$

Turning back to the general form of $W^-(t,p)^t$ and $H(t,p)^t$, the transformation is obtained applying to each pair of terms I'_k, H'_j such that $H'_j \neq I'_k$ and $H'_j \cap I'_k \neq \phi$ the following rule:

$\gamma'_j H'_j \rightarrow \gamma'_j(H'_j \cap I'_k) + \gamma'_j(H'_j \ominus I'_k)$ $\lambda'_k I'_k \rightarrow \lambda'_k(H'_j \cap I'_k) + \lambda'_k(I'_k \ominus H'_j)$

The bounds are built as follow:

- $\forall I_j$ not appearing in $H(p,t)^t$ there will be a bound $[\lambda_j, \infty)$
- $\forall H_j$ not appearing in $W^-(p,t)^t$ there will be a bound $[0, \gamma_j)$
- $\forall I_j$ appearing also in $H(p,t)^t$ with coefficient γ there will be a bound $[\lambda_j, \gamma)$

The computational complexity to obtain the expression of $\mathcal{LE}(t,p)$ may be combinatorial, depending on the number of terms of $W^-(t,p)^t$ and $H(t,p)^t$ and on the size of the tuples. However the calculus has to be performed only once at the net level for each place p and for each transition t connected to p via an input/inhibitor arc. This information may be saved as structural information of a given model. It is observable that however it does not depend on the size of the underlying PN because no unfolding is so far computed.

Refining Set $\mathcal{LE}(t,p)$

Each integer bound $[\alpha_k, \beta_k)$ associated to a term T_k of $\mathcal{LE}(t,p)$ determines whether instances $T_k(c)$ are potentially in $EN(\mathbf{m}, t)$: precisely, it happens if the number of tokens of colour c in place p is in the range $[\alpha_k, \beta_k)$. The set obtained is a super-set of the actually enabled instances. Actually $EN(\mathbf{m}, t)$ due to place p only, is given by the following Difference (3):

$$\Big\{ \bigcup_{c \in \mathcal{C}(p), k}^{\mathbf{m}(p)(c) \in [\alpha_k, \beta_k)} T_k(c) \Big\} / \Big\{ \bigcup_{c \in \mathcal{C}(p), k}^{\mathbf{m}(p)(c) \notin [\alpha_k, \beta_k)} T_k(c) \Big\} \quad (3)$$

As concern the running example the interpretation is: $\mathcal{LE}(t, p_1)$ provides a pattern for enabled colour instances of t with respect to p_1, if a colour instance c' of t matches one term T_k in the formal sum, and the multiplicity of a coloured token c in p_1 such that $c' \in T_k(c)$ ranges onto $[\alpha_k, \beta_k)$ then c' is potentially enabled. If instead the multiplicity of c in p_1 is out of the range, then c' is not enabled. For instance, the first term of $\mathcal{LE}(t, p_1)$ says that no instance (c_1, c_1), $\forall c_1 \in C_1$ of t will be ever enabled. The second and third terms say instead that the enabling of an instance (c_1, c_2), $c_2 \neq c_1$ requires that colour c_1 occurs at least once but no more than twice, while c_2 occurs at least twice. Now, assume $C_1 = \{a, b, c, d, e\}$, and consider the marking $\mathbf{m}(p_1) = 3.a + 2.b + 1.c$. According to Difference (3) $EN(t, p_1)$ due to place p_1 only is:

$$\big\{\langle b, S - b\rangle + \langle S - a, a\rangle + \langle S - b, b\rangle + \langle c, S - c\rangle\big\} / \big\{\langle a, S - a\rangle + \langle S - c, c\rangle + \textstyle\sum_{\forall x \notin \overline{\mathbf{m}(p_1)}}(\langle x, S - x\rangle + \langle S - x, x\rangle))\big\} = \{\langle b, a\rangle, \langle c, a\rangle, \langle c, b\rangle\}$$

Efficient Calculation of $EN(\mathbf{m}, t)$

To compute $E(\mathbf{m}, t)$ locally to place p, Difference (3) is not efficient because it considers all colours in $\mathcal{C}(p)$ domain. Depending on the colour domain size this computation could be enough expensive. Actually the computation of $EN(\mathbf{m}, t)$ may be efficiently done referring only to colours in $\mathbf{m}(p)$ and without regard to their multiplicity. The remainder of the section illustrates the procedure to obatain the enabled instances $EN(\mathbf{m}, t)$ using $\mathcal{LE}(t,p)$.

When no guards occur on $W^-(t,p)$, the formal expression $\mathcal{LE}(t,p)$ may be manipulated to efficiently compute $EN(\mathbf{m}, t)$. If this condition is met then $W^-(t,p)$ applied to any colour c of its domain returns a constant number of tokens, let it be K: $|\overline{W^-(t,p)(c)}| = K, \forall c \in \mathcal{C}(t),\ K \in \mathbb{N}$. For each colour $c \in \overline{\mathbf{m}(p)}$ it is considered the expression resulting form the application of $\mathcal{LE}(t,p)$ to c. Terms whose bounds are not satisfied by the multiplicity of c in $\mathbf{m}(p)$ do not compare in the expression. For shortness of notation let $\mathcal{EN}(c)$ be such expression, that is $\mathcal{EN}(c) = \mathcal{LE}(t,p)(c)$. The instances of t requiring K tokens from p, corresponding to the multiset elements having multiplicity K, are those *actually enabled* and are computed by intersecting such terms in $\mathcal{EN}$ whose lower bounds sum to K. In the running example the sum $\mathcal{EN}(a)$, $\mathcal{EN}(b)$ and $\mathcal{EN}(c)$ are respectively:

$$\overbrace{\underbrace{\langle S - a, a\rangle}_{[2,.)}}^{\mathcal{EN}(a)} \quad \overbrace{\underbrace{\langle b, S - b\rangle}_{[1,.)} + \underbrace{\langle S - b, b\rangle}_{[2,.)}}^{\mathcal{EN}(b)} \quad \overbrace{\underbrace{\langle c, S - c\rangle}_{[1,.)}}^{\mathcal{EN}(c)}$$

where the expression shows the lower bounds for each term with the purpose to point out the required instances from place p_1. This expression highlights the contributions due to the different colours a, b and c, occurring on place p_1. Because $K = |\overline{\langle X_1 + 2.X_2\rangle}| = 3$ the local enabled colour instances of t are obtained intersecting:

1) $\mathcal{EN}(a) \cap \mathcal{EN}_1(b) = \langle C_1 - a, a\rangle \cap \langle b, C_1 - b\rangle \rightarrow (b, a)$

2) $\mathcal{EN}(a) \cap \mathcal{EN}(c) = \langle C_1 - a, a\rangle \cap \langle c, C_1 - c\rangle \rightarrow (c, a)$

3) $\mathcal{EN}(b) \cap \mathcal{EN}_2(b) = \langle C_1 - b, b\rangle \cap \langle c, C_1 - c\rangle \rightarrow (c, b)$

Observe that since $\mathcal{EN}(b)$ is composed by disjoined terms it is useless intersect $\mathcal{EN}_1(b)$ and $\mathcal{EN}_2(b)$ even if their multiplicity sums to 3. Although the above expressions are not functional they are similar to those of the framework language $\mathcal{L}$, thus the same rewriting rules of the framework can be used to symbolically solve the intersections.

Observing steps $1), 2), 3)$ above, it can be noticed that the calculus actually may be done on functions rather than on instances of functions. If x_1, x_2 are variables on C_1 then steps $1), 2), 3)$ may be folded into a functional expression intersecting tuples in $\mathcal{LE}(t, p_1)$ whose lower bounds sum to constant K and using variables in place of colours:

$$\langle x_1, S - x_1\rangle \cap \langle S - x_2, x_2\rangle \rightarrow \langle x_1 \cap S - x_1, S - x_2 \cap x_2\rangle$$

which after the simplifications is:

$$\mathcal{LE}(t, p_1) = \langle x_1, x_2\rangle[x_1 \neq x_2]$$

with the following variables constraints for any marking $\mathbf{m}'$:

$$\mathbf{m}'(p_1)(x_1) \in [1, 3) \text{ and } \mathbf{m}'(p_1)(x_2) \in [2, \infty)$$

The last expression for $\mathcal{LE}(t, p_1)$ is marking independent, and it is computed at the net level and associated to the model. $\mathcal{LE}(t, p_1)$ is next utilised to compute $EN(\mathbf{m}, t)$ in a given marking, instancing the function $\langle x_1, x_2\rangle[x_1 \neq x_2]$ in the marking and checking the variables constraints. The interpretation of $\langle x_1, x_2\rangle[x_1 \neq x_2]$ says that the enabled instances of t in any marking $\mathbf{m}'$ are those (x_1, x_2) for which there are a number of x_1 tokens comprised in $[1, 3)$ and at least 2 tokens of colour x_2. From a practical point of view the enabled instances are enumerated considering each pair of colour in p_1 at $\mathbf{m}$ and checking the variables constraints: doing so the previous set is obtained.

Dealing with Guarded Functions

Arc functions having guarded tuples may map to multisets of different sizes, depending on the colour instance they are applied to. A guard in fact annuls a tuple for some colour instance of the transition colour domain: if $W^-(t, p)(c') = \phi$ then instance c' always satisfies the enabling condition due to the input arc constraints. Analogously, the constraints given by the inhibitor arc are satisfied by those colour instances of t annulling $H(t, p)$.

If $W^-(t, p)$ is null for some instance c', that is $W^-(t, p)(c') = \phi$, then, due to the transpose definition, it can be verified that $W^-(t, p)^t(c)$, $\forall c \in \mathcal{C}(p)$ never maps to a multiset containing c'. In the expression of $\mathcal{LE}(t, p)$ thus only terms T_k having bound $[0, n_k)$ are such that $T_k(c)(c') > 0$ for some c, whereas the other terms never map into a multiset containing c'. Thus, testing the enabling of instance c' of t only depends by the evaluation of the $H(t, p)$.

In Fig. 2 there are illustrated two variants for $W^-(t, p_1)$ respectively given by cases 1) and 2). In case 1) tuple $\langle X_1 + 2.X_2\rangle$ is guarded by $[X_1 \neq !X_2]$ which means that $W^-(t, p_1)$ is annulled by any colour instance $(c_1, c_2) \in C_1 \times C_1$ of t such that $c_1 = !c_2$. Whereas the inhibitor function never maps into ϕ. $\mathcal{LE}(t, p_1)$ results into the following expression, where $!^{-1}$ is the predecessor function:

$$\underbrace{\langle X_1, X_1\rangle}_{[3,3)} + \underbrace{[X_1 \neq !X_2]\langle X_1, S - X_1\rangle}_{[1,3)} +$$
$$+ \underbrace{[X_1 \neq !X_2]\langle S - X_1, X_1\rangle}_{[2,\infty)} + \underbrace{\langle X_1, !^{-1}X_1\rangle}_{[0,3)}$$

It can be verified in case 1) $EN(\mathbf{m}, t)$ due to place p_1 is:

$$\{(b, a), (c, a), (c, b), (d, c), (e, d)\}$$

To calculate such set the method requires some further step and refinement. A calculus analogous to the one of previous section leads to have as $\mathcal{LE}(t, p_1)$ in marking $\mathbf{m}(p_1) = 3.a + 2.b + 1.c$ the set $\{(c, a)\}$, in fact the functional form to compute the enabled instances from the marking is the same with the exception of the filter, $[x_1 \neq !x_2]\langle x_1, x_2\rangle[x_1 \neq x_2], x_1 \in [1, 3),\ x_2 \in [2, \infty)$. However, such result is partial because it does not take into account such instances of t the guard in the input arc function $W^-(t, p_1)$ maps to the empty multiset. An empty multiset means that those instances do not require any tokens from p_1, that is, they are always enabled with regard to place p_1. It is necessary to add to the above set all the instances annulling the input arc, these are symbolically expressed as $[X_1 = !X_2]\langle C_1, C_1\rangle$.

Due to the lower bound value, tuple whose bound is $[0, 3)$ when applied to instance c' of t always maps into multiset of tokens that just inhibits instance c'. In marking $\mathbf{m}$ the only colour with multiplicity that does not satisfy the upper bound is a thus it is necessary to subtract instance (a, e) from the previous set. The form of the guard annulling the input function may be always inferred by means of equivalent function rewriting.

In case 2) of Fig. 2 the input function $W^-(t, p_1) = X_1[X_1 \neq !X_2] + 2.X_2$, due to the guard, maps into multisets with different cardinality for different colour of transition t. It is possible to rewrite such expression into a sum of pairwise disjoint guarded terms $I_1[g_1] + I_2[g_2] + \ldots I_n[g_n]$ such that each I_k is a function mapping into multisets having a constant cardinality. In the example it is possible to write 2) as:

$$\underbrace{\langle X_1 + 2.X_2\rangle}_{I_1}[X_1 \neq !X_2] + \underbrace{2.\langle X_2\rangle}_{I_2}[X_1 = !X_2]$$

where $|I_1| = 3$ and $|I_2| = 2$. After such rewriting the algorithm proceeds computing for each term the local enabling functions $\mathcal{LE}_{I_1}(t, p_1)$ and $\mathcal{LE}_{I_2}(t, p_1)$ as previously described. These are respectively:

$$\overset{[3,3)}{\underbrace{\langle X_1, X_1\rangle}} + \overset{[1,3)}{\underbrace{\langle X_1, (S - X_1) \cap (S - !^{-1}X_1)\rangle}} +$$
$$+ \underbrace{\langle (S - X_1) \cap (S - !X_1), X_1\rangle}_{[2,\infty)} + \underbrace{\langle X_1, !^{-1}X_1\rangle}_{[0,3)}$$

and

$$\underbrace{\langle !X_1, X_1\rangle}_{[2,\infty)} + \underbrace{\langle X_1, S\rangle}_{[0,3)}$$

The locally enabled set results from subtracting the sum of partial minuends computed after step 2 from the sum of potentially enabled sets built during step 1.

$$\underbrace{2.\langle S-a\cap S-b,a\rangle}_{\mathcal{EN}(a)}\ \underbrace{1.\langle c,S-c\cap S-b\rangle}_{\mathcal{EN}(c)}$$
$$\underbrace{1.\langle b,S-b\cap S-a\rangle+2.\langle S-b\cap S-c,b\rangle}_{\mathcal{EN}(b)}$$

and

$$\underbrace{2.\langle b,a\rangle}_{\mathcal{EN}(a)}\ \underbrace{2.\langle c,b\rangle}_{\mathcal{EN}(b)}$$

from which using intersection the potentially enabled sets are respectively $\{(c,a)\}$ and $\{(b,a),(c,b)\}$, corresponding to the multiset terms of multiplicity 3 and 2, respectively.

The algorithm can be generalised in case the input and inhibitor arc functions of t are simultaneously annulled by some transition colour instance.

Compatibility with SWN symbolic marking

The most interesting feature of SWN is the representation of the system's symmetries by means of a structured syntax for both arc functions and colour domains; exploiting such *language* to denotes symmetries, very efficient algorithms have been utilised to build an aggregated state space, called Symbolic Reachability Graph (SRG), where markings are represented in a symbolic notation which allows to fold many ordinary markings into a single symbolic one (SM) - the ones so far treated are all ordinary markings. The SWN underlying stochastic process that use SM is a *lumped* CTMC, and also simulation runs may take advantages on such compact representation. These methods are described in literature as Symbolic Marking and Symbolic Firing Rule (Chiola et al., 1997).

A SM provides a syntactical equivalence relation on ordinary markings: two markings belong to the same SM iff they can be obtained from one another by means of a *permutation* on unordered colour classes that preserves static subclasses, and a *rotation* on ordered colour classes. A given SM is formally represented in terms of *dynamic subclasses*, that define a *parametric partition* of static subclasses.

The basic formula for local enabled instances $\mathcal{LE}(t,p)$ that the entire approach to symbolic enabling test relies upon, can be used to compute more efficiently the set of enabled symbolic colour instances of a transition in a given SM. The only difference is that it applies to multisets of dynamic subclasses instead of multisets of basic colours.

Referring to the running example of Fig. 2, the marking below is the representation of a SM respective of place $p1$:

$$3Z_1\ +\ 2Z_2\ +\ 1Z_3\ ,|Z_1|=|Z_2|=1,\ |Z_3|=2$$

Symbol Z_i denotes a dynamic subset, called subclass in SWN terminology, of basic colour class C_1: in other words it is a parametrical set of colours in C_1 with a given cardinality. An example of ordinary marking belonging to this SM is $\mathbf{m}'(p_1):3a+2b+1c+1d$.

Applying expression $\mathcal{LE}(t,p_1)$ to the above SM the method results in the symbolic multiset:

$$\underbrace{2.\langle C_1-Z_1,Z_1\rangle}_{\mathcal{EN}(Z_1)}\ \underbrace{1.\langle Z_2,C_1-Z_2\rangle+2.\langle C_1-Z_2,Z_2\rangle}_{\mathcal{EN}(Z_2)}$$

$$\underbrace{1.\langle Z_3,C_1\rangle}_{\mathcal{EN}(Z_3)}$$

where it has been used the following step: $(S-X_1)(Z_3)\rightarrow C_1$. The enabled symbolic colour instances of t due to place p_1 in the SM considered are thus expressed as tuples of dynamic subclasses and correspond to the multiset terms of multiplicity 3, obtained by intersecting the pairwise disjoint terms whose coefficients sum to 3, that is:

$$\{\langle Z_2,Z_1\rangle,\langle Z_3,Z_1\rangle,\langle Z_3,Z_2\rangle\}$$

The second and third terms above denote subsets of $C_1\times C_1$ whose cardinality is 2 whereas the first term denote sets composed by a single instance in $C_1\times C_1$.

An Example from the Literature

This section shows the application of the algorithm to an example taken from the literature (Ballarini et al., 2003), where a fault-tolerant distributed memory algorithm is modelled and analysed through SWN. The Distributed Memory (DM) is a scalable multi-task component based on memory replication that, using a voting and task synchronization algorithms, implements safe read/write operations on application variables. Due to the complexity of the entire model, simulation was used within the *GreatSPN* package to obtain the performance analysis.

Figure 3 depicts a part of the entire SWN model, it represents two operations done by a master-task during its lifecycle: the check of variable replicas after a distributed voting (transition $CHECK\_COMP$) and the issuing of a read-request received from an application task on a given DM variable (transition RD_MST). Basic colour classes respectively denote: C_1 the task identities, C_2 the application variables, C_3 the values assumed by variables, C_4 the control code sent by DM tasks during execution (e.g. rd_ok which means a succeeded read operation). Some colour classes are partitioned in static subclasses: in particular $C_{1,1}$ denotes the DM task group, and $C_{4,1}$ the *read ok* signal.

The arc functions of the selected sub-model can be treated neither using the heuristics defined in (Gaeta, 1996) nor the approach defined in (Jean-Michel Ilie and Omar Rojas, 1993): in particular, the presence of complement functions, repeated projection symbols and guards avoid exploitation of any known optimisation.

Assuming $C_{1,1}=\{dm_1,dm_2,dm_3\}$, $C_2=\{var_1,var_2,var_3\}$, $C_3=\{v_1,v_2,v_3\}$, as first example of the algorithm application it is considered the enabling

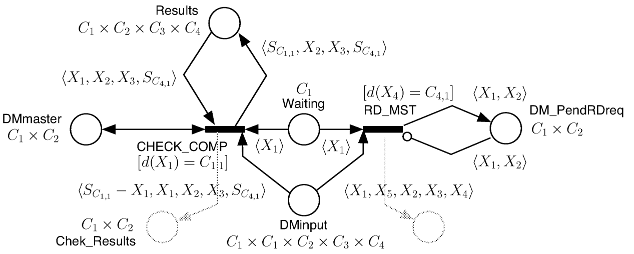

Results
$C_1\times C_2\times C_3\times C_4$
$\langle S_{C_{1,1}},X_2,X_3,S_{C_{4,1}}\rangle$
$\langle X_1,X_2,X_3,S_{C_{4,1}}\rangle$
C_1
Waiting
$[d(X_4)=C_{4,1}]$
RD_MST
$\langle X_1,X_2\rangle$
DM_PendRDreq
$C_1\times C_2$
DMmaster
$C_1\times C_2$
CHECK_COMP
$[d(X_1)=C_{1,1}]$
$\langle X_1\rangle$
$\langle X_1\rangle$
$\langle X_1,X_2\rangle$
$\langle S_{C_{1,1}}-X_1,X_1,X_2,X_3,S_{C_{4,1}}\rangle$
$\langle X_1,X_5,X_2,X_3,X_4\rangle$
$C_1\times C_2$
Chek_Results
DMinput
$C_1\times C_1\times C_2\times C_3\times C_4$

Figure 3: An Example from the Literature

of transition $CHECK\_COMP$, whose colour domain is $C_1 \times C_2 \times C_3$, in the marking below:

$\mathbf{m}(DMinput)$:
$\langle dm_1, dm_2, var_1, v_1, rd\_ok\rangle + \langle dm_3, dm_2, var_1, v_1, rd\_ok\rangle +$
$+\langle dm_1, dm_2, var_2, v_2, rd\_ok\rangle$
$\mathbf{m}(DMmaster) : \langle dm_2, var_1\rangle + \langle dm_2, var_2\rangle + \langle dm_3, var_3\rangle$
$\mathbf{m}(Results) : \langle dm_2, var_1, v_1, rd\_ok\rangle + \langle dm_2, var_2, v_2, rd\_ok\rangle$
$\mathbf{m}(Waiting) : \langle dm_2\rangle$
$\mathbf{m}(DM\_PendRDreq) : \langle dm_3, var_2\rangle$

Such marking represents a situation in which a read request on var_1, whose master is dm_2, has been successfully managed by the remaining DM tasks, while a read request on var_2 has been managed only by dm_1. Applying the algorithm $\mathcal{LE}(CHECK\_COMP, DMinput)$ is:

$$\overbrace{\langle X_2, X_3, X_4\rangle}^{[1,\infty)} \circ$$
$$\circ[X_1 \in C_{1,1} \wedge X_2 \in C_{1,1} \wedge X_5 \in C_{4,1} \wedge X_1 \neq X_2]$$

and $EN(\mathbf{m}, CHECK\_COMP)$ due to place $DMinput$:

$$2.\langle dm_2, var_1, v_1\rangle \; + \; 1.\langle dm_2, var_2, v_2\rangle$$

The term occurring with multiplicity 2, which is the cardinality of any application of the input function, represents the locally enabled colour instance of transition $CHECK\_COMP$.

As second example, let us consider transition RD_MST, and place $DM\_PendRDreq$. The local enabling expression turns out to be:

$$\underbrace{\langle X_1, S, X_2, S, S_{C_{4,1}}\rangle}_{[0,0]}$$

Applying the algorithm, the locally enabled set is:

$$\langle C_1, C_1, C_2, C_3, C_4\rangle - \langle dm_3, S, var_2, C_3, C_{4,1}\rangle$$

which finally results in:

$$\langle C_1 - dm_3, C_1, C_2, C_3, C_4\rangle + \langle dm_3, C_1, C_2 - var_2, C_3, C_4\rangle +$$
$$+\langle dm_3, c_1, var_2, C_3, C_4 - C_{4,1}\rangle$$

CONCLUSIONS

This paper illustrates how the framework for SWN structural analysis (Capra et al., 2005) is a valuable basis for the implementation of an efficient and general enabling test in SWN. The algorithm illustrated is symbolic because it manipulates the functions labelling the input and inhibitor arcs by means of the framework's rewriting rules. Compared to previous works (Gaeta, 1996) and (Jean-Michel Ilie and Omar Rojas, 1993) the algorithm is supposed to be more efficient as it computes a functional form for $\mathcal{LE}(t, p)$ only once at the net level. $\mathcal{LE}(t, p)$ tries to exploit as much as possible the SWN structural information so to both move most of the computation at the net level and to alleviate $EN(\mathbf{m}, t)$ task at the marking level, when colour instances are considered. For this reason a consistent reduction of the computational overhead is expected. Current and future works are in two directions: 1) Produce an implementation of the algorithm, once this step is achieved it is planned its integration into the *greatSPN* tool (Chiola et al., 1995) and the verification of how simulation runs take advantage in terms of overhead reduction in the event-list management; 2) Utilising the framework for SWN, enhance the enabling test here presented exploiting structural relations of the net, such as the structural conflict and the structural causal connection (Evangelista and Pradat-Peyre, 2004).

REFERENCES

Ballarini, P., Capra, L., De Pierro, M., Franceschinis, G., Jun. 2003. "Memory Fault-tolerance Mechanisms: Design and Configuration Support through SWN Models." In *Proceeding of the 3rd International Conference on Application of Concurrency to System Design* (ACSD'03). IEEE. Guimaraes, Por.

Capra, L., De Pierro, M., Franceschinis, G., June 2005. "A High Level Language for Structural Relations in Well-Formed Nets." In *Proceedings of the 26th International Conference on Application and Theory of Petri Nets.* Miami, USA. LNCS 3536, pp. 168–187, Springer-Verlag.

Chiola, G., Dutheillet, C., Franceschinis, G., Haddad, S., 1993. "Stochastic Well-formed Coloured Nets for Symmetric Modelling Applications." IEEE TC 42(11) (11), 1343–1360.

Chiola, G., Dutheillet, C., Franceschinis, G., Haddad, S., 1997. "A Symbolic Reachability Graph for Coloured Petri Nets." Theoretical Computer Science B (Logic, semantics and theory of programming) 176 (1&2), 39–65.

Chiola, G., Franceschinis, G., Gaeta, R., Ribaudo, M., 1995. "GreatSPN-1.7 – graphical editor and analyzer for timed and stochastic Petri nets." In *Performance Evaluation*, Vol. 24, No. 1-2, pp.47–68, 1995.

Chiola, G., Gaeta, R., October 1995. "Efficient simulation of SWN models." In *Proceedings of the Sixth International Workshop on Petri Nets and Performance Models*, pp. 137-146. IEEE.

Evangelista, S., Pradat-Peyre, J. F., October 2004. "An Efficient Algorithm for the Enabling Test of Colored Petri Nets." In *Proceedings of the Fifth Workshop and Tutorial on Practical Use of Coloured Petri Nets and the CPN Tools*, Kurt Jensen (Ed.), pp. 137–156.

Gaeta, R., September 1996. "Efficient Discrete-event Simulation of Colored Petri Nets." In *IEEE Transactions on Software Engineering* Vol.22, No.9, pp. 629–639.

Jean-Michel Ilie, Omar Rojas, June 1993. "On Well-formed Nets and Optimizations in Enabling Test." In *Proceedings of the 14th International Conference on Application and Theory of Petri Nets*, LNCS Vol. 691, pp. 300–318, Springer-Verlag.

Jensen, K., 1997. "Coloured Petri Nets. Basic Concepts, Analysis Methods and Practical Use." Volume 1, Basic Concepts. Monographs in Theoretical Computer Science, Springer-Verlag.

Jensen, K., Rozenberg, G. (Eds.), 1991. "High-Level Petri Nets: Theory and Application." Springer Verlag.

ESA_PetriNet tool: Extraction Scenarios & Analyzer by Petri Net model Application to the extraction of feared scenarios in a landing gears system

Malika Medjoudj, Hamid Demmou, Robert Valette

mmedjoudj@gmail.com, hamid@laas.fr, robert@laas.fr

ABSTRACT

The evaluation of the dynamic reliability of the computer-controlled systems (mechatronic systems, flight computers, etc.) constitutes one of the major concerns of industrials in the reliability field, whereas exists limited number of tools. The objective of this paper is to present a new tool ESA_PetriNet (Extraction Scenarios & Analyzer by Petri Net model) developed in the aim of extracting critical scenarios from a Petri Net model of the system. The continuous dynamics of the system is taken into account by temporal abstractions. The effectiveness of this tool is illustrated on a real industrial example.

Key words: Reliability, computer-controlled systems, dynamic reliability, checking of properties, hybrid system, Petri net, Linear Logic.

1. Introduction

We present in this paper a first prototype of the ESA_PetriNet tool (Extraction Scenarios & Analyzer by Petri Net model), which makes it possible the extraction of feared scenarios from a Petri Net model with respect of the continuous aspect of the system. We present the method and the basic of the algorithm of this tool in section 2. The selected case study will be presented in section 3 with the Petri Net modelling. The ESA_PetriNet prototype, its use to generate the feared scenarios and the discussion of the results will be presented respectively in sections 4, 5 and 6. Section 7 will be devoted to the adaptation of this tool to the checking of temporal properties and we end this work by a conclusion.

The application of our method requires the modelling of the system by a Predicates Transitions Differential Stochastic Petri Net model. A preliminary analysis will refine the fields of the variables according to various accessible markings by reasoning on the invariants of places, the differential equations associated to the places and the thresholds associated to the transitions. The invariants of places are used to determine the possible dynamics, and which places can be simultaneously marked when a given place is marked. Then a temporal abstraction is made by associating to the transitions a temporal interval of firing corresponding to the time, which the system can spend to reach the state, in question.

2. Method of extraction of the scenarios in hybrid system

2.1. Principles

This method is made up of two steps [Medjoudj et al. 04]: a backward reasoning and a forward reasoning. The backward reasoning takes as an initial marking in the reversed Petri net model, the only target state (feared) and seeks exhaustively all the scenarios making it possible to consume the initial marking (feared state since we reason forward) and reach a final marking composed only of places associated to the normal operation. The forward reasoning takes as an initial state these places of normal operation in the initial Petri net model. Its objective is to locate the junctions between the feared behaviour and the normal operation of the system as well as the conditions implied in these junctions. Thus we have not only the explanation of the dangerous behaviour but also of strategies allowing its avoidance. A significant point of the method is that the context in which occurred the feared event is enriched gradually. Each scenario is given in form of a partial order between the events necessary to the appearance of the feared event what differs from a failure tree, which gives a whole of static combinations of the partial states necessary for obtaining the feared state.

2.2. Dealing with continuous dynamics by temporal abstractions

The method that we propose takes into account the conditions associated to the firing of certain transitions. These conditions are thresholds involving continuous variables. By temporal approximation of the hybrid dynamics, these thresholds are transformed to durations, which correspond to time that the system puts to reach when the transitions are enabled. As, for the moment we are working only from a qualitative point of view and not from a quantitative point of view, determining the firing order of the transition only interests us. Thus we can have a situation where two transitions t1 and t2 are enabled if we consider the ordinary Petri net, but whose are such as t1 will be always fired before t2 if we consider the temporal abstraction. In the generation of the scenarios only the firing of t1 will be considered since that of t2 before t1 would be in fact incoherent with the continuous dynamics. This appears in the form of a priority: if t1 and t2 are enabled, only the case of t1, priority, is examined.

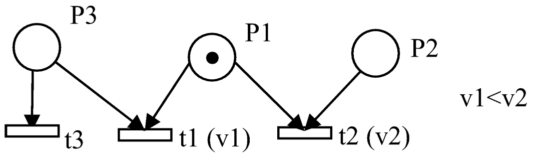


Figure1. Priority due to the thresholds of transitions

The taking into account of these precedence relations coming from the continuous dynamics and not specified by the ordinary Petri net allows to reduce the number of scenarios generated by eliminating a certain number of incoherent scenarios with respect to continuous dynamics.

Let us consider an example. In figure1 we suppose that the differential-algebra system associated to the place P1 guarantees that the variable x is increasing. We associate to the transition t1 the threshold x=v1 and to the transition t2 the threshold x=v2 with v1<v2. Finally, we suppose that when the token arrives in the place P1 we have always x< v1. So, if the place P3 is marked, the transition t1 will be fired before t2 since the threshold associated to t1 is lower than that of t2. In this case we don't consider the scenario associated to the firing of t2. On the other hand, if t3 is already fired (place P3 empty) and if the place P2 is marked, t1 cannot be fired and then t2 will be fired.

2.3. Precedence and direct and indirect causality

In the example above, we will finally examine only one type of scenarios, those for which the transition t2 is fired after t3. So, we have a precedence relation between the firing of t3, which empties the place P3 and that of t2, however there is no place connecting t3 to t2. This precedence relation is so a consequence of continuous dynamics and thresholds associated to transitions t1 and t2. We speak in this case about indirect precedence relation and about indirect causality. The direct precedence relations and causality are those that are highlighted by the only Petri net, i.e. by the only discrete aspect.

2.4. The algorithm

Thanks to a temporal abstraction of continuous dynamics, we highlight relations of indirect precedence and causality between certain transitions firing. In the algorithm, this is expressed in the form of rules of priority (a certain transition is not fired if another is enabled). In the expression of the results, i.e. scenarios, that appears in the form of indirect precedence/causality relations between transitions that are not connected by a place. So we will restrict, sometimes strongly, the number of generated scenarios and for each scenario the unit of the firing sequences is coherent with the strict partial order relation associated to the scenario. The fact that these scenarios are coherent with respect to dynamics continuous will simplify the phase of checking. The steps of the algorithm are detailed in [Medjoudj 06]. We note that only one execution of the algorithm generates automatically several scenarios.

Indeed starting by a feared state, we go up in the chain of causalities in the reversed Petri net until reaching a normal operating condition. So, we stop the backward reasoning and we start a forward reasoning. In the latter, we start with the normal operating condition found previously, and we animate the original Petri net to reach the feared state by enriching marking when it is necessary. All the possible and coherent scenarios with respect to the continuous dynamics of the system are generated.

3. Case study

3.1. General description

This case study concerns a Rafale landing gears system of Dassault Aviation. This system is a modified and simplified version of a benchmark proposed by the French working group STRQDS [STRQDS 02], which has been first presented by [Boniol and Carcenac 02] and first studied by [Villani et al. 03] in its hybrid version. The aim is to apply our approach to extract the feared scenarios.

This system is composed of three landing gears (a nose train, a left train and a right train), which must be in retracted position for flying (fast) and in extended position for landing. These three trains are interdependent. Each train is made up with:

- A box to fix the gear in retracted position.
- A door, which has to be open before extending or retracting a gear and automatically closed when the movement is complete.

System operating

The gears are controlled by means of a three-position command: E, R and B

- When the command is E, the box doors are opened, the three gears are extended and the doors are closed.
- When the command is R, the box doors are opened, the three gears are retracted and the doors are closed.
- In the intermediary position, the command is B, and the gears are blocked in their current positions.

Feared events

We are interested only to the feared events related to the extending of the gears. We have three feared events for each train:

- None-opening of the box door to extend the gear,
- None-extending of the gear,
- None-closing of the box door after extension of the gear.

The feared events related to the retracting of the gears can be obtained in a similar way. In this paper, we don't take into account the feared events related to the command B. We do not describe the command B as well as the feared events, which are related to it.

3.2. Composition of the system

The system is made up of a computer which controls three sets gear/door. According to the position of the gears, doors (provided by the sensors) and of the state of the plane the computer establishes the order to be produced on the system via the hydraulic actuating cylinders. Figure2 includes the various components of the system. It is composed of:

A computer which sends the command E for extending gears, the command R for retracting gears and a backup command which allows only the extending gear in case of blocking in extending. To order the extending or the retracting of the gears, an UP/DOWN pallet is provided to the pilot: UP (command E) and DOWN (command R).

Three sets gear/door actuated by hydraulic actuator cylinders. The cylinders are moved by electro-valves by setting pressure. Each actuator cylinders of a door is ordered by an electro-valve for opening and an electro-valve for closing. Each gear is ordered by an electro-valve for extending and an electro-valve for retracting. The gears actuators have two functionalities: extending/retracting and locking in low position.

Four backup electro-valves to free a landing gear at extending in case of failure of a specified electro-valve (blocking of actuator). The first backup electro-valve replaces one of the three electro-valves of opening of the

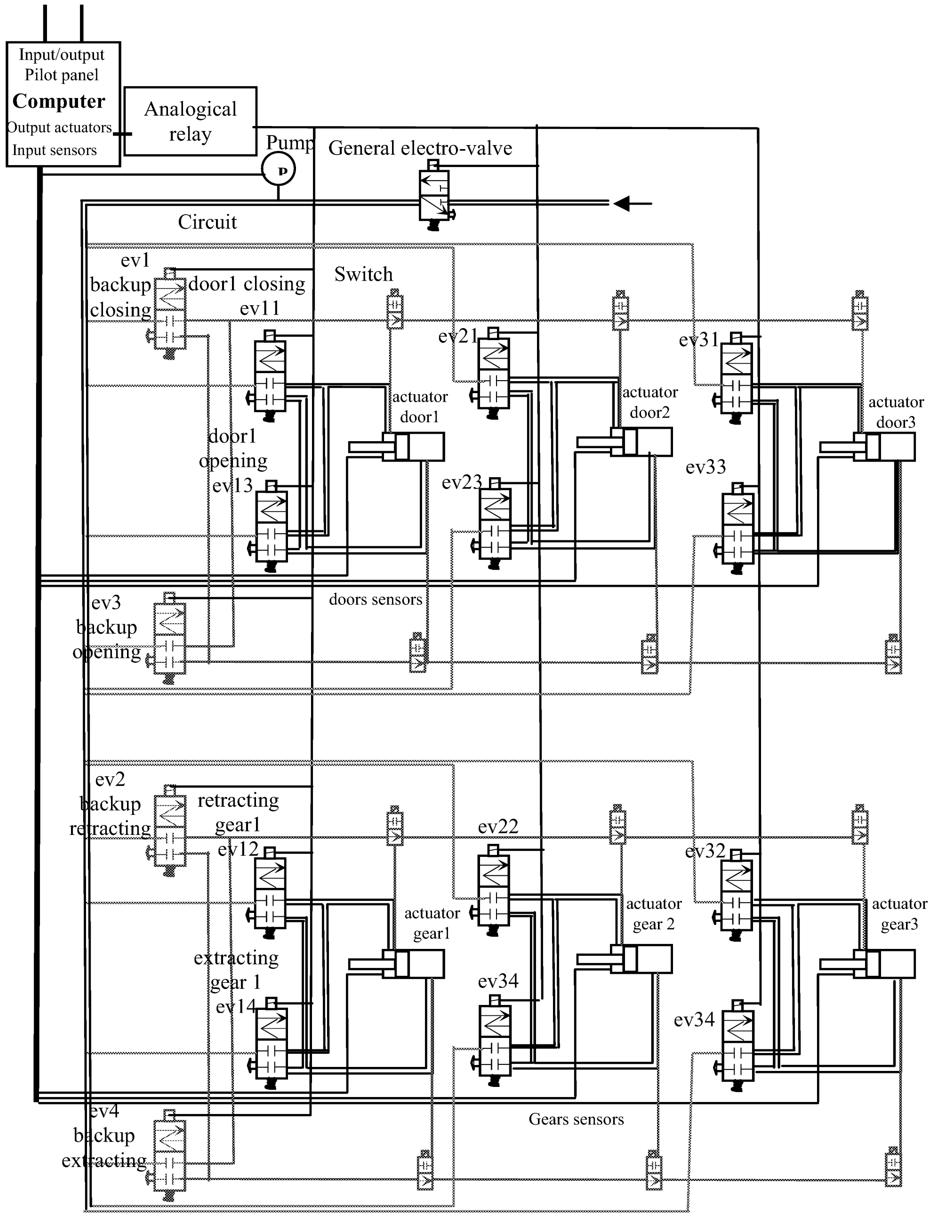


Figure2. Components of the landing system

three doors. The second backup electro-valve replaces one of the three electro-valves of extending gears. The third backup electro-valve replaces one of the three electro-valves of closing doors and the fourth is used in case of blocking of the gears while retracting. We assume that, the backup electro-valve can be used only for one actuator at the same time. The switches on figure2 correspond to electro-valves, which can be opened or closed according to the commands of the computer. The computer orders the opening of an electro-valves (switch) to set pressure on the actuators of a door i (gear i) and orders closing the others so that the backup electro-valves will be used only by one door (gear) at the same time (to give maximum pressure).

A general electro-valve that sets pressure to the circuit.

Position sensors indicate position of gears and doors.

An analogical relay isolates the computer from the electro-valves. The relay remains open a certain time after the last change made in the command component (UP/DOWN handle). This time is supposed to be sufficient for opening or retracting the landing system. When the pilot acts again on the UP/DOWN handle, the relay is closed.

3.3. Modelling

A global view of the Petri net model is given in figure3. Place p_1 represents the state in which the three gears are retracted. Transition t_1 is fired when the command E is issued. This event generates three concurrent connects, one for each gear. Transition t_2 is fired when the three gears are extended. The right part of the figure3 corresponds to the retracting command. Note that the block between pi0 and pi4 corresponds to Petri net model.

3.3.1. Petri net model for the extending

To simplify the model, we assume that the computer, the pump, the general electro-valve and the sensors provide their function and are not failing. So they will not be modelled. In this example we consider only the failures corresponding to the different electro-valves.

A Simplified view of the Petri net model for extending gears is given in figure4. Note that the places ps_{i0} and p_{i4} are the same as in figure3. The model contains three identical nets, which follow one another: 1) opening of the three doors, 2) extending of the three gears, 3) closing of the three doors.

We extract the feared scenarios in a modular way. As we mentioned, a feared state is the consequence of the one of the following events: 1) not-opening of a door i for extending gear i, 2) not-extending gear i, 3) not-closing of a door i after the extending gear i.

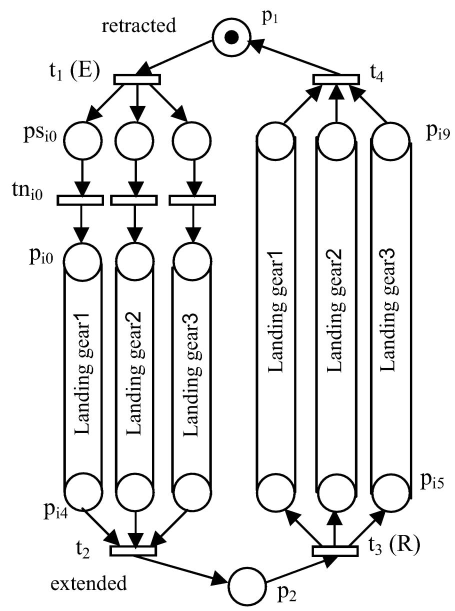


Figure3. General view of the Petri net

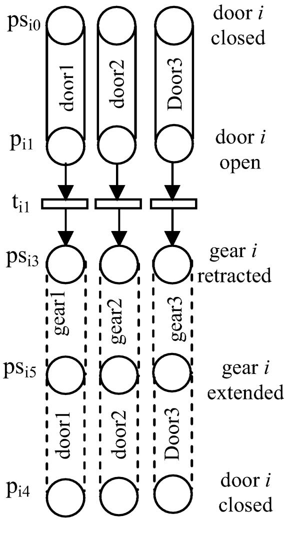


Figure4. Simplified view of door opening

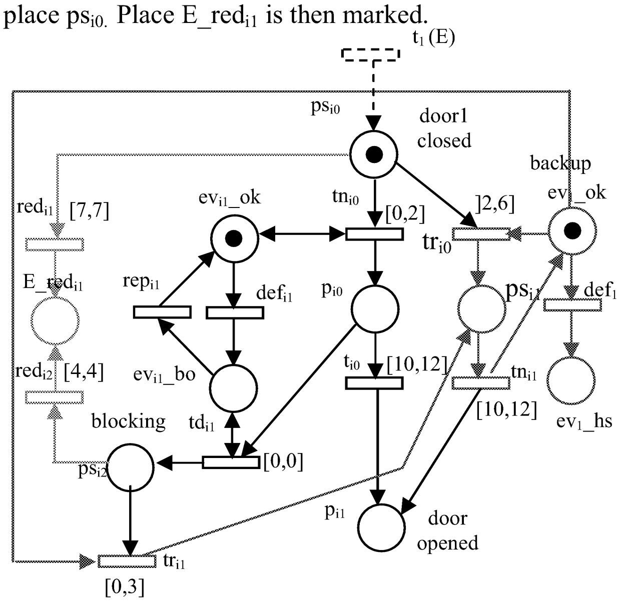
We will focus on the feared scenarios corresponding to not opening of a door i for extending gear i. So, we consider the first Petri net corresponding to the opening of the three doors. These three doors are interdependent since they use the same backup electro-valve that can be used only by one door at the same time.

3.3.2. Petri net model for opening a door with failure

The door (gear) position is determined by the position of the corresponding hydraulic actuating cylinder. The failure (blocking of a door or a gear) corresponds to the failure of the electro-valve that sets pressure to the corresponding actuator.

Transition tn_{i0} in figure5, represents the request for opening of the ev_{i1} electro-valve. This electro-valve can be available (place ev_{i1} _ ok is marked) or failing by firing transition def_{i1}. If ev_{i1} is available, transition tn_{i0} is firing and place p_{i0} is marked. Place p_{i0} corresponds to the opening of the door. Its continuous dynamics is taken into account by a temporal abstraction. The minimal duration for the door opening is 10 seconds and its maximum duration is 12 seconds. Time interval corresponding to the temporal constraint is represented by [10, 12] associated to transition t_{i0}. As there are no other entry places for transition t_{i0}, this means that the token must remain in the place p_{i0} at least 10 seconds and at most 12 seconds before firing (place p_{i1} corresponding to the position of door open is marked).

The time response of the ev_{i1} electro-valve is 2 seconds. That is represented by the temporal interval [0, 2] associated to transition tn_{i0}. This means that ev_{i1} is failing if there is no movement of the opening door i after the excitation of the door opening during two seconds. So, the computer requires the use of the backup electro-valve ev_1 with a time response of 3 seconds represented by the temporal interval]2, 6] associated to transition tr_{i0}. If this electro-valve is available, transition tr_{i0} is fired and place ps_{i1} is marked (reconfiguration of the system). As there is no other transition, which can consume it, this token cannot remain in ps_{i1} more than 12 seconds. The backup electro-valve is then released (place ev_1_ok marked) by firing transition $tn_{i1.}$ If the ev_{i1} electro-valve is failing and the backup electro-valve is out of service or is used by one of the two other doors, so the first transition red_{i1} driving towards the feared state is fired 7 seconds after marking of place $ps_{i0.}$ Place E_red_{i1} is then marked.



Response time ev_{i1}: 2 seconds
Response time $ev_{1:}$ 3 seconds
Door opening: 10 seconds

Figure5. Petri net model for opening door i

Firing the immediate transition td_{i1} because of the failure of ev_{i1} electro-valve can interrupt the dynamics of the place p_{i0}. This corresponds to a blocking of the door in opening represented by the marking of place ps_{i2}. In this case the backup electro-valve ev_1 is requested. If it is available and is not used by one of the two other doors, transition tr_{i1} corresponding to a reconfiguration after blocking is fired. If not the second transition red_{i2} driving towards the feared state is fired. We assume that the electro-valves ev_{i1} can be repaired after failure by firing transition rep_{i1} and if the backup electro-valve ev_1 is failing, it is out of service. The transitions related to the

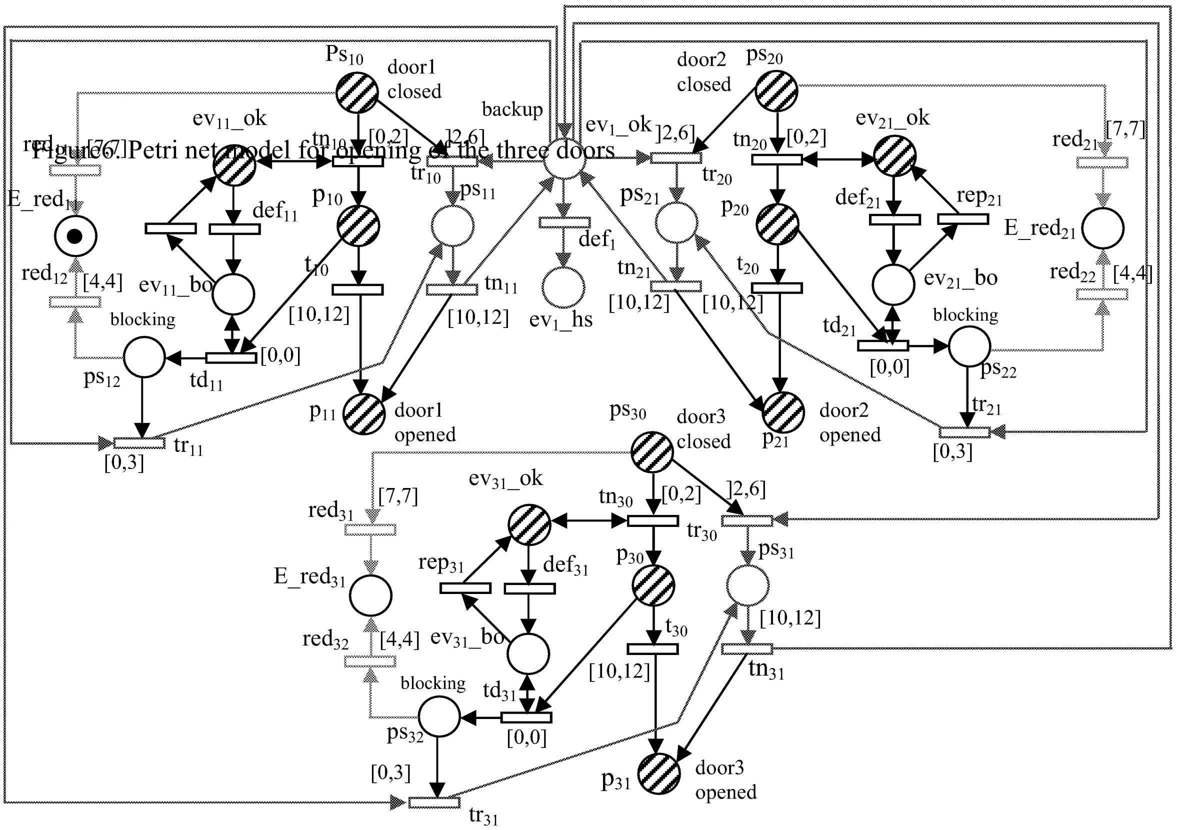


Figure6. Petri net model for opening of the three doors

Figure6. Petri net model for opening of the three doors

failures and repairs of the electro-valves are stochastic and can be fired at any moment. The temporal abstraction of this model is a t-temporal Petri net modelling the priority transitions firing.

3.3.3. Petri net model for opening of the three doors

The striped places in figure6, modelling the opening of the three doors, represent normal operating. We will seek the feared scenarios corresponding to not-opening the door1, i.e. all the scenarios which lead to the marking of the place E_red_{11}. The feared scenarios corresponding to not-opening the door2 and door3 can be obtained in a similar way. The following invariants of the places are obtained by a structural analysis with TINA tool [Berthomieu et al. 03]:

$M(11\_bo) + M(ev11\_ok) = 1$
$M(ev21\_bo) + M(ev21\_ok) = 1$
$M(ev31\_bo) + (ev31\_ok) = 1$
$M(E\_red11) + M(p10) + M(p11) + M(ps10) + M(ps11) + M(ps12) = 1$
$M(E\_red21) + M(p20) + M(p21) + M(ps20) + M(ps21) + M(ps22) = 1$
$M(ev1\_hs) + M(ev1\_ok) + M(ps11) + M(ps21) + M(ps31) = 1$ $M(E\_red31) + M(p30) + M(p31) + M(ps30) + M(ps31) + M(ps32) = 1$

4. ESA_PetriNet prototype

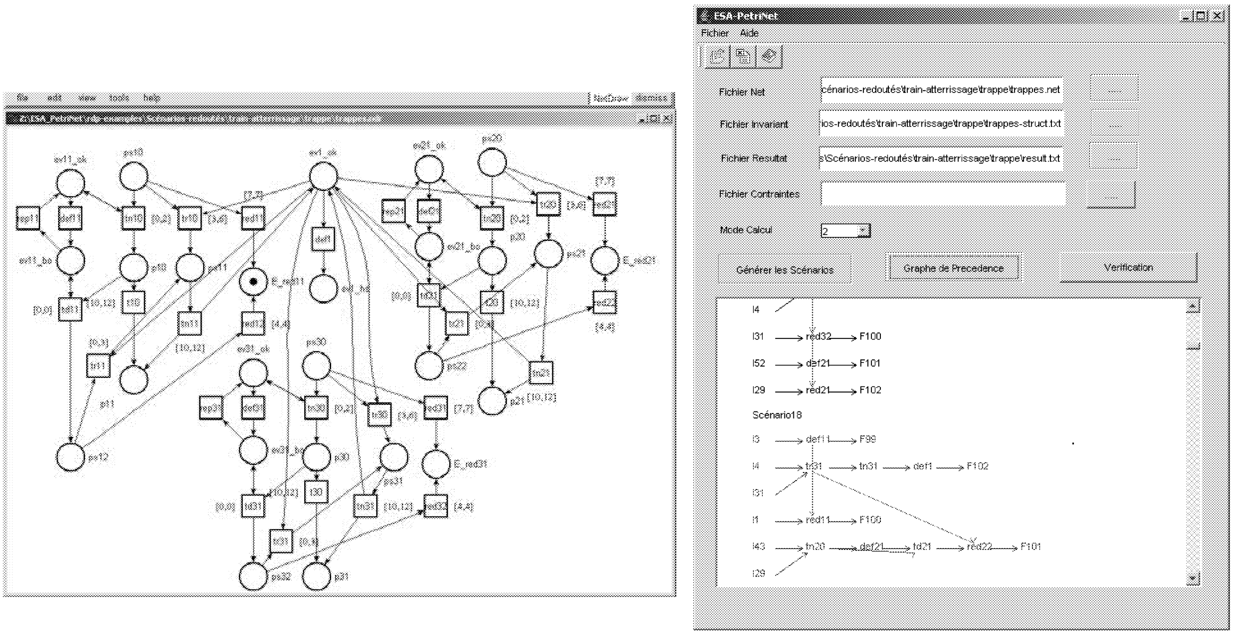
We chose Java for the development of ESA_PetriNet to have a better portability of the tool. So, we can use this software on various material platforms and under various operating systems. We have interfaced ESA_PetriNet with TINA tool (Time Petri Net Analyzer) [Berthomieu et al. 03]. Although this tool is dedicated to the ordinary Petri nets and the t-temporal Petri nets and not to Petri nets associated with differentials-algebra equations, TINA tool has several advantages. First of all it is possible to use its graphic editor to describe our Petri nets. Then, although the temporal abstraction more direct of a Predicates Transitions Differential Stochastic Petri Net model is obtained in the form of an arc-pt-temporal Petri net, it is sometimes possible, to transform it in a t-temporal Petri net. ESA_PetriNet tool is made up mainly of the following functions:

Input files: the input files of the tool correspond to a textual description of the t-temporal Petri net model of the studied system. The current version of ESA_PetriNet considers the output files of TINA (Time Petri Net Analyzer) version 2.7.4. We can use easily its graphic interface to model our system by Petri nets. For checking temporal constraints, an additional file containing the temporal constraints is introduced.

Operating mode: extraction of feared scenarios for reliability needs a backward and a forward reasoning (mode 2). The generation of scenarios for checking temporal constraints is done only by a simple backward research (mode 1).

Extraction of scenarios: it is the principal function. After analysis of the input files, the tool extracts the necessary data structure for the algorithm.

Recording results: generated scenarios are memorized in a textual file. We extract all the scenarios (normal operation, reconfigurations and feared scenarios) because we need to have precise information concerning the dynamics of the system.



Fichier Net: cénarios-redoutés\train-atterrissage\trappe\trappes.net

Fichier Invariant: ios-redoutés\train-atterrissage\trappe\trappes-struct.txt

Fichier Resultat: s\Scénarios-redoutés\train-atterrissage\trappe\result.txt

Fichier Contraintes:

Mode Calcul: 2

TINA ⟶ Input files ⟶ ESA_PetriNet ⟶ Scenarios

Figure7. Screen shots of ESA_PetriNet

Precedence graph: we chose precedence graph to present the generated scenarios. Direct and indirect causality Relations are illustrated in different colour.

Checking of temporal constraints: this function uses the temporal constraints introduced by the user. It checks the satisfaction of the constraint by all scenarios. The treatment stops with the violation of a constraint and a counterexample is then generated. This function is used in simple cases and it has to be supported by addition of other algorithm.

5. Use of ESA_PetriNet to generate feared scenarios in the case of study

A general view of ESA_PetriNEt is given in figure7. After edition of Petri net of the system on the graphic editor of TINA, we generate two input files: file.net, which is a descriptive file of the Petri net (RdP) and file-struxt.txt for the invariant of places. We introduce these two files in the graphic interface of ESA_PetriNet and we give a name for the results file. Then we click on the button "generate scenarios" and the button "precedence graph" to illustrate these scenarios in the form of graph of precedence. The operating mode in this case is 2.

ESA_PetriNet generates all the scenarios bringing tokens in place E_red11. With the version of the algorithm that supports the continuous dynamics we obtain 28 scenarios in which 21 are feared. With the version that doesn't support continuous dynamics we have 114 scenarios in which 87 are feared. In this example we did not examine the loops, we choose to put the transitions rep11, rep21, rep31 in the list of a forbidden transition transitions to avoid the loop problem.

Among the 21 feared scenarios generated by taking into account the continuous aspect of the system, we find those corresponding to:

- Failure of an electro-valve ev_{i1} (def_{i1}) and the failure of the backup electro-valve ev_1 (def_1): sc1, sc2, sc3, sc4, sc5, sc10, sc12.
- Failure of an electro-valve ev_{i1} (def_{i1}) and the use of the backup electro-valve ev_1 (def_1) by another door (tr_{i0} or tr_{i1}): sc6, sc7, sc8, sc9, sc11, sc13, sc17, sc19, sc21, sc22, sc23, sc25, sc27, sc28.

To facilitate the identification of the feared scenarios among the scenarios of normal operating and reconfiguration, we chose to illustrate them with different colour. The scenarios are recorded in the form of a series of events couple (e1, e2). The direct causality relation between e1 and e2 is noted by e1&e2 and the indirect causality relation is noted by e1#e2. In the graphic interface of ESA_PetriNet, the direct and indirect precedence relations are represented with different colours.

Example: Sc1: I3&def11 I4&def1 I9&red32 I10&red22 I1&red11 def11&F99 def1&F100 red32&F101 red22&F102 red11&F103 def1#red32 def1#red22 def11#red11 def1#red11. Ii correspond to initial events and Fi to final events.

6. Discussion

6.1. Impact of the continuous aspect

In the case of the scenarios generated by taking into account the continuous aspect of the system, the order of appearance of the events related to continuous dynamics is taken into account. For example the scenario sc13: {I1, **def11**, **tr21**, red32, **red11**, tn21, def1} presented by the precedence graph in figure 8, the cause of the feared event (transition red11) is the failure of the electro-valve corresponding to the door1 (transition def11) and the use of the backup electro-valve by the door2 (transition tr21). The dotted arrows between def11, red11 and tr21, red11 represent the indirect causality relations.

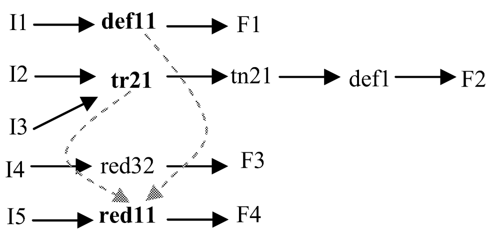


Figure8. Precedence graph of scenario sc13

6.2. Without the continuous aspect

Figure 9 and 10 represent two cases of scenario without taking into account the continuous aspect of the system. In the scenario Sc37: {I1, **def11**, **red11**, red32, **tr21**, tn21, def1}, the order of the events related to continuous dynamics is not taken into account. The scenario Sc73: {I1, def11, red11, def31, red22} is non admissible. Taking into account the continuous aspect of the system eliminates this kind of scenarios.

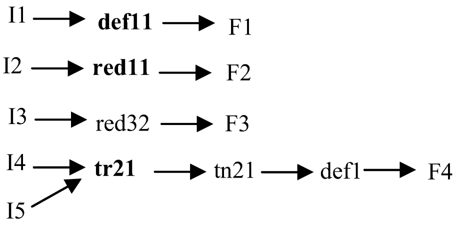


Figure9. Precedence graph of scenario sc37

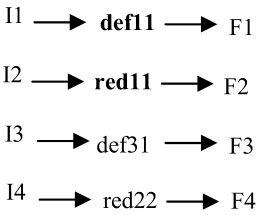


Figure10. Precedence graph of scenario sc73

Generation of scenarios in the whole model of extracting gears

If one of the places E_red_{i1}, E_red_{i2}, E_red_{i3} is marked we have a feared event. In figure11, a new feared state E_red is defined by gathering the three preceding ones. We seek all the scenarios, which lead to the marking of this place, i.e. the firing of the one of the transitions t_red_{i1}, t_red_{i2} or t_red_{i3}. The detailed Petri net of extracting of the landing gears is given in [Medjoudj 06].

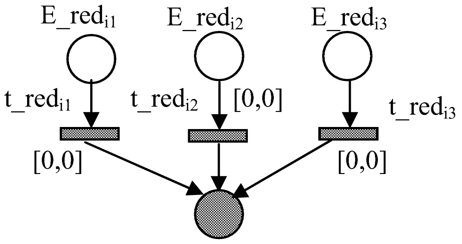


Figure11. Feared state for extracting of the 3 landing gears

7. Use of ESA_PetriNet for checking

Checking consists in determining if the system satisfies certain properties. We have adapted ESA_PetriNet tool developed to extract feared scenarios in the hybrid systems to checking of properties. The adaptation consists in adding a function of checking once all the scenarios are generated. The properties that we want to check are temporal. We have used ESA_PetriNet to check the maximum duration of a scenario and duration between two orders.

7.1. Principle

Checking impose simply to give the whole of scenarios. For example to show that a state is reached at a certain time with a certain set of command, it is necessary to give all the compatible scenarios with this command and leading to the desired state. We need only the backward reasoning. The forward reasoning having to be used only to check that the set considered command solves all the conflicts and when the command is sent, the system will necessarily will evolve according to one of the obtained scenarios.

A scenario is a partial order between events. If we want a quantitative checking, i.e. if we want to show that for a set of fixed value, the limit is lower than a certain value, so, the temporal abstraction of the hybrid system allows the calculation of the maximum duration of each scenario and check if the property is checked.

The step consists in generating all the scenarios then the temporal abstraction is used to express the quantitative temporal constraints associated to each scenario. A temporal constraints network is associated to each scenario. Finally the constraints on the parameters that allow the checking of the property are deduced. The important point is that model-checkers generate sequences of events, whereas a scenario encapsulates many sequences and allows a symbolic reasoning on the temporal constraints.

7.2. Case study

We have used ESA_PetriNet to generate scenarios from hybrid system model corresponding to a physical system with a command in case of the flight computer [Medjoudj 06]. The goal was to check that the duration between two commands sent by two different computers is limited by the interval [dmin, dmax]. It means that two commands can never be sent simultaneously. The list of forbidden transitions is used to take into account the effect of the command on the system. A simple back exploration is enough to generate all the scenarios leading to the target state. Then a temporal abstraction is used to obtain temporal constraints networks. Once the temporal constraints network is built, calculations of the temporal constraints associated to the longest ways is typical techniques used in artificial intelligence. The base is the traditional algorithm of Floyd-Warshall [Floyd 62] [Warshall 62].

We have also used ESA_PetriNet in the landing gears example to check that the duration of a scenario is lower than certain limit [Medjoudj 06], [River et al 05]. In simple cases like this, it is not necessary to assign numerical values to the parameters. Instead of simply showing that the property is true or false for a set of given value of the parameters, we are able in this case to give the constraints that must check the parameters so that the property is true.

8. Conclusion

We have presented in this paper, a first prototype of the ESA_PetriNet tool "Extraction & Scenarios Analyzer by Petri Net model" which allows extraction of feared scenarios from a Petri net model by taking into account the continuous aspect of the system. This tool can be also used to check certain temporal properties of the system like the duration of a scenario or accessibility between two states.

We have validated this tool by applying it to an example of a consequent industrial size, conceived from a real system (a Rafale landing gears system of Dassault Aviation). We have deliberately restricted the example to be clear. The Petri net of the simplified example contains nearly a hundred of places. By using the symmetry of the system, it is possible to constitute the total model in a relatively easy way. The generated scenarios are coherent with respect to the continuous dynamics of the system and we noted that the number of eliminated scenarios for their incoherence with respect to continuous dynamics is significant. The computing time takes few seconds. We have also applied this tool for checking certain temporal properties concerning flight computers and the landing gears system. The problem of this last is similar to the one of the availability of production.
Among the prospects for the improvement of the tool, there is the taking into account the minimality of the scenarios, to eliminate the unnecessary events and the quantification of theses scenarios by a Monte Carlo simulation [Kalos et al 86]. The checking part can be supported by algorithms of research longest ways.

Acknowledgement: Ayda Saidane, Doctor Paul Sabatier, for his councils of programming.

9. Bibliography

[Berthomieu et al 03] B. Berthomieu, P.O. Ribet, F. Vernadat, «L'outil TINA Construction d'espaces d'états abstraits pour les réseaux de Petri et réseaux Temporels», Modélisation des Systèmes Réactifs, MSR'2003, Hermes, 2003.

[Boniol et Carcenac 02] F. Boniol et F. Carcenac, « Une étude de cas pour la vérification formelle de propriétés temporelles » Journées FAC, Toulouse, France, 26-26 March 2002.

[Floyd 62] Robert W Floyd, «Algorithm 97: Shortest path», Communications of the ACM Vol.5 Issue 6, (June 1962), page.345.

[Kalos et al 86] M.H Kalos et Whitlock P.A, «Mont Carlo methods», Volume I: basics, John Wiley and Sons, New York, 1986.

[Medjoudj 06] M. Medjoudj, «Contribution à l'analyse des systèmes pilotés par calculateurs: Extraction de scénarios redoutés et vérification de contraintes temporelles». Thèse doctorale de l'Université Paul Sabatier, Mars 2006, Toulouse.

[Medjoudj et al 04] M. Medjoudj, S.Khalfaoui, H. Demmou, R. Valette, «A method for deriving feared scenarios in hybrid systems», Probabilistic Safety Assessment and Management (PSAM'7 - ESREL'04), Berlin (Allemagne), 14-18 June 2004.

[Rivière et al 05] N. Riviere, H. Demmou, R. Valette, M. Medjoudj, «Symbolic temporal constraint analysis, an approach for verifying hybrid systems», 16th IFAC World Congress, Prague (République Tchèque), 3-8 July 2005.

[STRQDS 02] STRQDS, «Présentations d'études de cas », Octobre 2002. http://www.laas.fr/strqds.

[Villani et al 03] E. Villani, J.C. Pascal, P.E. Miyagi and R. Valette, « Differential predicate transition Petri nets and objects, an aid for proving properties in hybrid systems », ADHS 03, IFAC Conference on Analysis and Design of Hybrid Systems, p. 117–122, Saint-Malo, France, 16-18 June 2003.

[Warshall 62] Stephen Warshall, «A theorem on Boolean matrices». Journal of the ACM Vol.9 Issue .1, pages 11-12, January 1962

INTRODUCTION TO COMPLEX SYSTEMS SIMULATION

Holistic Metrics, a Trial on Interpreting Complex Systems

J. Manuel Feliz-Teixeira
António E. S. Carvalho Brito

April 2006

GEIN – Section of Management and Industrial Engineering
Faculty of Engineering of the University of Porto, PORTUGAL
Emails: feliz@fe.up.pt; acbrito@fe.up.pt

KEYWORDS: Complex system, holistic metrics, time domain, frequency domain, base states, state projection.

ABSTRACT

In this article is proposed a simple method for estimating or characterize the behaviour of *complex systems*, in particular when these are being studied throughout simulation. Usual ways of treating the complex output data obtained from the activity (real or simulated) of such a kind of systems, which in many cases people classify and analyse along the *time domain*, usually the most complex perspective, is herein substituted by the idea of representing such data in the *frequency domain*, somehow like what is commonly done in *Fourier Analysis* and in *Quantum Mechanics*. This is expected to give the analyst a more holistic perspective on the system's behaviour, as well as letting him/her choose almost freely the *complex states* in which such behaviour is to be *projected*. We hope this will lead to simpler processes in characterizing complex systems.

1. Introduction

There are presently very few notes on the kind of metrics that could be reliable and of practical relevance when applied to the interpretation of complex systems behaviour. These systems are often based on intricate structures where a high number of entities interact with each other. Metrics are there for appropriately characterizing the nodes or individual parts of such structures, or small groups of them, but when the intent is a measure for the complete structure either they fail or appear to be too simplistic. That is certainly a good reason for modelling those cases using a *strategic* point of view, removing the *time* variable from the process, as in doing so the complexity is reduced *a priori*.

But when a dynamic and detailed representation is essential, the interpretation of the results and the characterization of the system frequently fail. This issue seems sometimes also related to a certain tendency impregnated in the minds to look at the systems from a pre-established perspective. At this point, however, perhaps this may be considered a conflict between different scientific approaches: the classical western reductionism, of anglo-saxonic inspiration, which believes the best approach is to break the system into small parts and understand, model or control those parts separately and then join them together, therefore looking at the world in an individualist way; and a more holistic approach, a vision slowly spreading and largely inspired by oriental cultures, which considers that each part of the system must be seen together with the whole and not in isolation, and therefore locates the tone in how the interactions between such parts contribute to the whole behaviour. Hopp & Spearman (2001, pp.16), for instance, comment about this saying that "*too much emphasis on individual components can lead to a loss of perspective for the overall system*".

A significant number of authors defend this opinion, pointing out the importance of developing a more holistic point of view to interpret and study systems behaviour, in a way that analyses maintain enough fidelity to the system as a whole. As

Tranouez et al. (2003), who apply simulation to ecosystems, would say: a complex system is more than the simple collection of its elements.

In management science, for instance, the "western" approach frequently generates difficulties at the *interfaces* between elements, typically of inventory or communication type. On the other hand, as *just-in-time* (JIT) systems give better emphasis to the relations and interactions and are continuously improving, the overall movements tend to be more harmonious. JIT already looks at systems in a certain *holistic* way. The same seems to be true in regard to other fields where simulation is applied, and mainly when the number of states to simulate is high.

2. Holist measuring (a proposal)

But, what concerning metrics? How can one measure such a high number of states typically found in complex systems in order to effectively retrieve from them some sort of useful information?

As a metric is a *characterization*, we could think that maybe the modern *Data Mining* (DM) techniques could be extensively applied, for instance. These techniques use decision trees and other algorithms to discover hidden patterns in huge amounts of data, and are nowadays applied to almost any problem based on extensive data records, for instance, in *e-Commerce* for customer profile monitoring, in genetics research, in fraud detection, credit risk analysis, etc., and even for suspected "terrorist" detection (see Edelstein, 2001; Edelstein, 2003). However, they often imply the usage of high performance computers, sometimes with parallel processors, as well as huge computational resources to analyse *GBytes* or even *TBytes* of data. They are useful when any single record of data can be precious for the future result, and thus when all data *must* be analysed.

On the other hand, in many practical simulations a significant amount of data is not significant for the final conclusions, the simulation process is in itself a filter, and therefore such data may well be ignored in the outputs, even if it could have been essential to ensure the detailed simulation process to run. In the perspective of the author, maybe there is a way that could deserve some attention: the idea is to filter such data during the simulation execution and, at the same time, to turn the measures probabilistic by using an approach somehow inspired by the *Fourier Analysis* and the *Quantum Mechanics*. That is, to represent the overall *system state* (ψ) in terms of certain *base functions* (ψ_i), and then to measure the *probabilities* (α_i) associated with each of these functions. The interesting aspect of this is that each base state function (ψ_i) could even be arbitrarily chosen by the analyst, and the probabilities (α_i) easily computed during the simulation process. Final results would then be summarised in some expression of the form:

$$\psi = \alpha_1 \psi_1 + \alpha_2 \psi_2 + \ldots \alpha_j \psi_j + \ldots \alpha_n \psi_n \quad (1)$$

which could be interpreted as: there is a probability of α_1 that the system will be found in the state ψ_1, a probability of α_2 that the system will be found in the state ψ_2, etc. This would be the final measure of the system, in a sort of characterization of expectations under certain conditions. This also corresponds to *projecting* the system behaviour into the generalised vectors base of *state functions* (ψ_i). The amounts α_i simply correspond to the values of those projections.

In *Fourier Analysis*, for instance, the complex behaviour observed in the *time axis* (see the example of figure 1) is substituted by the decomposition of such a signal into *sine* and *cosine* mathematical functions, and that way transferred to the *frequency* domain.

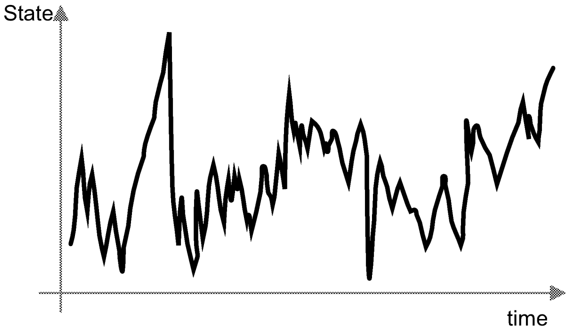


Fig. 1 Example of a general complex signal

The result is that the analyst is now much more able to visualize and to interpret the complexity of the previous signal, since it is as if this signal would be now expressed in terms of *patterns* (see example of figure 2). What firstly appeared as a confusing and almost randomly up-and-down behaviour may now be simply understood as the summation of some sinusoidal patterns with different amplitudes. *Quantum Mechanics* uses a similar formalism. We believe that the method proposed here will help generating such a clean view also when applied to the behaviour of *complex systems*.

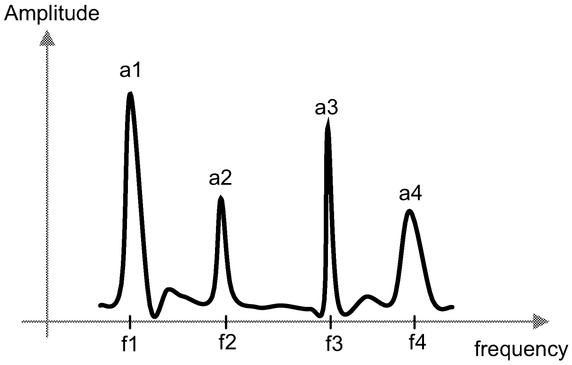


Fig. 2 Typical signal in the frequency domain

The present proposal may also be understood as an attempt to represent the system behaviour in terms of a sort of generalised histogram, where the *categories* are the functions ψ_i, which may correspond to the frequencies f_i in the previous figure, and the probabilities α_j are made to correspond to the amplitudes a_j in the same figure. In terms of this figure, the analyst would recognize a probability of a_1 that the system would be found in the state f_1, a probability of a_2 that the system would be found in the state f_2, etc.

3. An imaginary example

To help explain this, we can imagine a complex system like the Supply Chain shown in figure 3, for example.

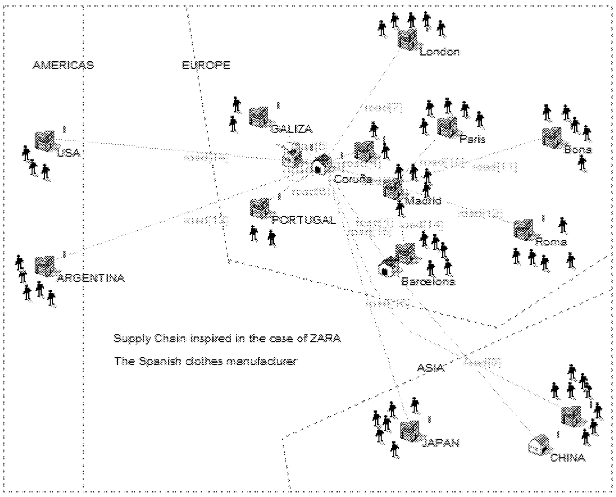


Fig. 3 Imaginary Supply Chain inspired by ZARA

This is an example inspired by the company ZARA, the trendy Spanish clothes manufacturer of La Coruña. This company, from the INDITEX group, is worldwide known as a paradigm of success, despite its owner, and major manager, Mr Ortega, the second richest person in Spain, refusing several conventional practices claimed by most schools of management. ZARA refuses, for instance, the idea of advertisement. Forgive me if indirectly I am advertising it here.

Returning to our subject, how can we apply our concept of holistic metrics to retrieve some useful information from such a complex case[1]? How can we specify the *base functions* (or *base states*) in which the system's behaviour will be *projected*? How will we calculate and represent the respective projections?

First of all, we have to choose the ψ_i functions into which the measures will be *projected*. We may choose them in terms of some specific *conditions* related to the information that must be obtained from the system. For example, if Mr Ortega is concerned about the levels of *stockouts*, *holding costs*, *service level*, *turnover*, etc., which are typical measures of Supply Chain Management, he may for example define some sort of base functions by using conditions of the type:

ψ_1 – *Stockouts above 7%*
ψ_2 – *Holding costs above 5%*
ψ_3 – *Service level under 75%*
ψ_4 – *Turnover under 2*

Then, while the system is running, it must be *projected* into these set of functions, that is, the occurrences of each of these conditions must be counted up, whenever they are true.

Supposing n_j the accumulated number of occurrences of the condition ψ_j, and N_j the total number of its samples, an estimation of α_j can simply be computed as:

$$\alpha_j = n_j / N_j \tag{2}$$

And the overall system state will therefore be expressed as:

$$\psi = (n_1/N_1)\psi_1 + (n_2/N_2)\psi_2 + (n_3/N_3)\psi_3 + (n_4/N_4)\psi_4 \tag{3}$$

Notice that, in general, *base functions* are

[1] In this figure is represented less than perhaps 10% of the real ZARA global Supply Chain structure.

chosen to be orthogonal, or independent of each other, but in fact that is not a *must* for using this type of representation. One can also *project* a system into *non orthogonal* axis. As we said previously, such a measure may be seen as a characterization of expectations under certain conditions. The overall system state is, in reality, represented by the following weighted equation:

$$\alpha_1 \textit{ (Stockouts above 7\%)} + \alpha_2 \textit{ (Holding costs above 5\%)} + \alpha_3 \textit{ (Service level under 75\%)} + \alpha_4 \textit{ (Turnover under 2)} \quad (4)$$

Now, if we build a histogram out of this data, we will *characterize* the system by means of a probabilistic graphical format, obtaining something of the type:

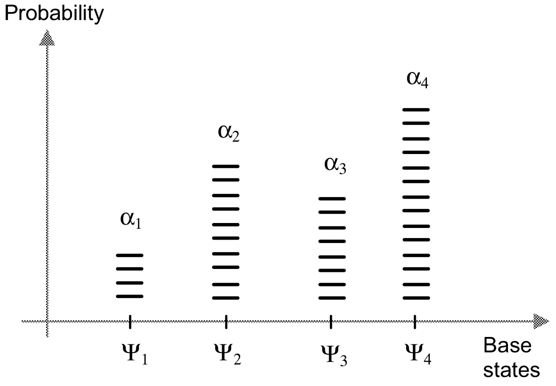


Fig. 4 Characterization of the system's behaviour

Were the probabilities are the α_i.

So, once the *base states* are well defined by the analyst, the characterization of the system is possible, no matters how complex the system is. We recall that in many practical cases the analyst is mainly focused in being sure that certain variables of the model do not cross some upper or lower limits, or, if they do, with which probability it happens.

In order to evaluate the system in a wider range of modes of behaviour, several studies of this kind can be made with the system operating in different conditions. That will make possible to improve the knowledge about the system, or its characterization.

The former example was taken from a typical Supply Chain problem (see Feliz-Teixeira, 2006, pp. 222), but this technique can be applied in general to other complex systems. For example, in a traffic system of a town, the *complex states* could be chosen to be the number of cars exceeding a certain value in a certain region, the travel time exceeding a certain value in another region, the number of public vehicles reaching a certain zone inferior to the minimum required, etc. As we recommend that these base functions (or *complex base states*) be well defined before simulation takes place, it implies that the simulation objectives must be well known prior to the start of the simulation process. Not always this is possible, of course, since simulation can be used to detect anomalous situations not predictable by means of other methods, for example.

This technique may, however, be also used as a method for analyse any sort of results, by being directly applied to the raw outputs of the complex system. In that case, the simulation will be a standard process and all the work is done by data manipulation. The results, in principle, will be the same, but that approach will in general be much more time consuming.

Finally, we would like to emphasise that we use the term "*holistic metric*" for distinguishing this kind of approach from those approaches which usually characterize systems by means of *averages* and *standard deviations* taken over a certain number of variables (usually a high number). These, as we know, frequently confuse the analyst's mind with the complexity of the results, instead of allowing a useful interpretation of the system's behaviour. Quantity of information is not all, and sometimes it can even generate confusion instead of clarity, if it is in excess. Besides, the method presented here goes on the trend of the "holistic" mind that seems to emerge in our days, as we defend.

4. CONCLUSIONS

Complex results generated by a *complex system* are very much dependent on how the analyst looks at the system and on how such results are analysed. We would say that any complex system can be minimally understood as long as the analyst knows what to search for, that is, if the objectives of the study are previously defined. This is because such objectives can in reality be used to establish the *base functions* (vectors) of an imaginary space where the complex behaviour will be *projected*, that way giving an automatic meaning to the results.

This may also be seen as an attempt to measure the outputs of systems in the *frequency domain* (as in *Fourier Analysis* and in *Quantum Mechanics*), instead of in the *time domain* where signals usually are more difficult to interpret. Although no practical cases have yet been studied based on the idea presented in this article, we expect to use and test this approach in our next studies of simulation. We would also be pleased with receiving some feedback from anyone who decided to apply the same logic.

References:

Edelstein, H. (2001). Pan For Gold In The Clickstream. *InformationWeek.com*.

Edelstein, H. (2003). Using Data Mining to Find Terrorists. *Data Mining Review*.

Feliz-Teixeira, J. M. (2006). *Flexible Supply Chain Simulation (thesis).* Faculty of Engineering of Universitry of Porto, Portugal.

Hopp, W. J., & Spearman, M. L. (2001). *Factory Physics* (second ed.): Irwin McGraw-Hill.

Tranouez, P., Lerebourg, S., et al. (2003). *Changing the Level of Description in Ecosystem Models: an Overview.* Paper presented at the The 2003 European Simulation and Modelling Conference, Naples, Italy.

Simulating Dynamic Behaviours in Complex Organisations: case study application of a well structured modelling approach

M Zhen and R H Weston
MSI Research Institute, Loughborough University
Loughborough, Leics., LE11 3TU, UK.
+44 1509 227514 / 227502 R.H.Weston@lboro.ac.uk

KEYWORDS

Manufacturing, decision support systems, dynamic modelling, combined simulation, process-oriented.

ABSTRACT

The paper reports a case study application of a systematic approach to modelling complex organisations, centred on simulation modelling. The approach leads to populated instances of complementary model types, in ways that systematically capture, validate and facilitate various uses of organisational understandings, knowledge and data normally distributed amongst multiple knowledge holders. An example enterprise model of a capacitor manufacturing company is illustrated as are derivative causal loop models that structure and enable the design and use of a general purpose simulation model.

DEFINING ORGANISATIONAL CONTEXTS: 'PROCESS MAPPING'

In general manufacturing organisations are very complex and need to be reconfigured during their useful lifetime. Typically many people need to be involved in designing and changing organisations. This is because (1) necessary understandings and knowledge about what can and cannot be done is normally distributed amongst many knowledge holders, (2) various personnel will normally have different responsibility for, and hence need to 'buy into', identified changes, and (3) because a range of business, managerial, technical and social skills are needed to realise organisational change on any significant scale.

It follows that creating simulation models of complex organisations, to support organisation design and change decision making, presents many difficulties. These difficulties mitigate against creating 'complete', 'accurate' and 'general purpose' simulation models that can be (a) usefully deployed by all relevant personnel and (b) updated as organisational and environmental changes occur. Consequently in general:

A) Structured simulation modelling methodologies are needed to create (individual and collective) simulation models such that (i) they represent specific realities of any target manufacturing organisation, (ii) simulated behaviours can be interpreted within the specific organisational context, (iii) the organisational context is appropriately decomposed into coherent but essentially decoupled organisation segments that can be usefully modelled and updated as ongoing organisational change occurs.

B) Multiple and coherent simulation models are needed, to encode all relevant organisational properties, knowledge and data that impact on the various decisions that need to be made about complex organisations;

Bearing in mind the set of requirements listed under A) and B), the present authors have studied the use of enterprise modelling techniques as a means of creating 'process maps' of target manufacturing organisations. The aim has been to evolve the basis of a modelling methodology which has potential to advance the use of simulation modelling in complex manufacturing environments.

The development of 'process maps' in manufacturing industries has become popular as a means of visualising what organisations 'do' (or 'should do in the future') so as to add value in the form of product and service generation for customers and profit or improved working conditions for stakeholders. In theory public domain enterprise modelling (EM) techniques can support 'process mapping'. In so doing that can (I) provide an ISO standard way of representing specific organisational contexts, and (II) break this specific context down into 'organisation segments' which are not too complex to model using current modelling technologies.

This paper describes how an ISO standard EM reference architecture and methodology (namely CIMOSA) has been used to: create a 'process map' of a case study manufacturing organization; decompose that map into segments of that organisation's reality that can usefully be modelled; structure the design, development and deployment of simulation models, based on the complementary use of causal loop models; facilitate the interpretation of simulated behaviours of segments of the reality, with respect to causal and temporal impacts that individually modelled organisational segments would have on overall business behaviours of that organisation.

CASE 'PROCESS MAPPING'

The case study company (referred to as CAP) is specialized in the development, production and sale of electrolytic capacitors and is an acknowledged world leader in the development of chip capacitors. It has overseas suppliers of raw material which it transforms into chip capacitor, epoxy-coated capacitor and non-solid electrolyte capacitor products for customers around the globe. Although CAP produces small strategic stocks of capacitor types (by forecasting market needs), primarily it makes capacitors to order. Figure 1 shows a so called context diagram of the 'make capacitors to order' process resourced by CAP. Here ISO standard CIMOSA (Computer Integrated Manufacturing Open Systems Architecture) enterprise modelling conventions (AMICE 1993) have been used.

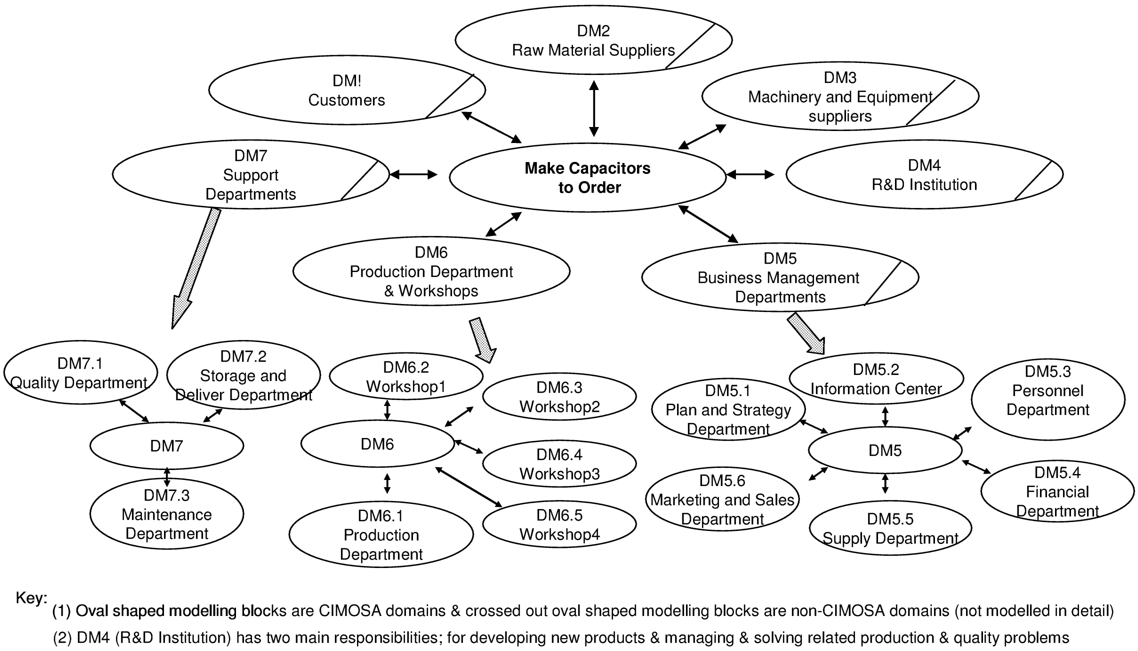

DM2
Raw Material Suppliers
DM!
Customers
DM3
Machinery and Equipment suppliers
DM7
Support Departments
Make Capacitors to Order
DM4
R&D Institution
DM6
Production Department & Workshops
DM5
Business Management Departments
DM7.1
Quality Department
DM7.2
Storage and Deliver Department
DM6.2
Workshop1
DM6.3
Workshop2
DM5.1
Plan and Strategy Department
DM5.2
Information Center
DM5.3
Personnel Department
DM7
DM6
DM6.4
Workshop3
DM5
DM5.4
Financial Department
DM7.3
Maintenance Department
DM6.1
Production Department
DM6.5
Workshop4
DM5.6
Marketing and Sales Department
DM5.5
Supply Department
Key:
(1) Oval shaped modelling blocks are CIMOSA domains & crossed out oval shaped modelling blocks are non-CIMOSA domains (not modelled in detail)
(2) DM4 (R&D Institution) has two main responsibilities; for developing new products & managing & solving related production & quality problems

Figure 1 Top level context diagram of CAP's making to order processes

This paper is focused on CAP's manufacturing business unit (DM6) because the company has experienced major bottle-neck production problems, making it difficult to reliably satisfy customer orders on time. Figure 2 shows the main event, information, material and resource interchanges between DP6 (the main Domain Process (DP) which DM6 owns) and other DPs by using a CIMOSA interaction diagram.

To further document the processes used by CAP, structure diagrams were employed to decompose each domain process into so called business processes and their ('atomic' building block) enterprise activities. Figure 3 shows a small part of the structure diagram created to explicitly document DP6 (the Produce Capacitors process segments) currently used by CAP.

CIMOSA structure diagrams were observed to provide a useful communication tool, which can align work processes to department responsibilities.

More detailed information about the flow of activities within process segments (and their related resource requirements and assignments, information flows and control flows) was described using many related CIMOSA activity diagrams; these diagrams flesh out processing requirements of each domain under consideration. In the case of CAP this was observed to require significant modelling effort but this effort was rewarded by formally documenting the current process network used by CAP; thereby externalising a significant body of understandings, knowledge and data about CAP in a standardised manner. Also it was observed that activity diagrams provide an

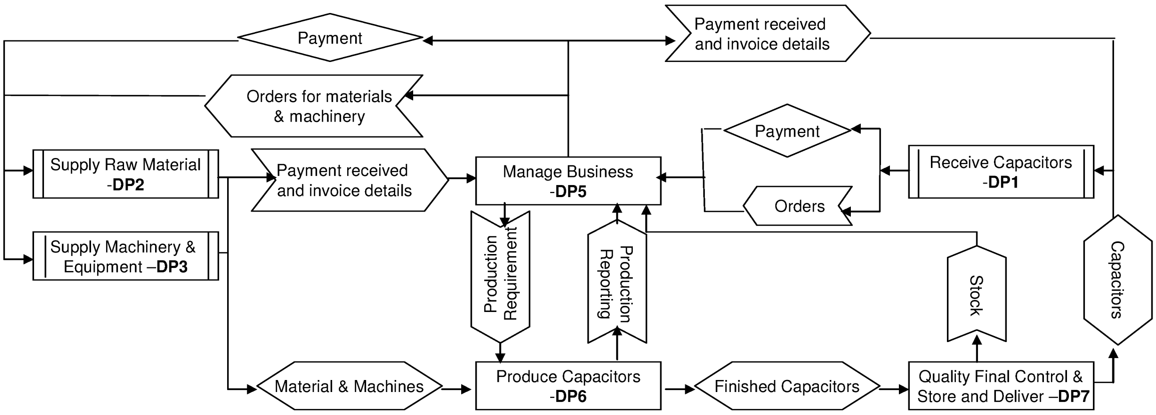


Figure 2 Top level interaction diagram of CAP's "make to order" domain processes

explicit and useful decomposition of processes into process segments that are essentially decoupled from each other, such that they can be modelled effectively using proprietary simulation techniques; hence in theory EMs can help understand and address CAP planning and scheduling problems as a precursor to selecting and implementing ERP or other proprietary management software. Figure 4(a) provides an illustrative activity diagram that documents, at a high level of abstraction, the make capacitor to order production process deployed by CAP; whilst Figure 4(b) shows how needed human and machine resources, and information and materials flows can be attributed to elemental activities that the BPs comprise.

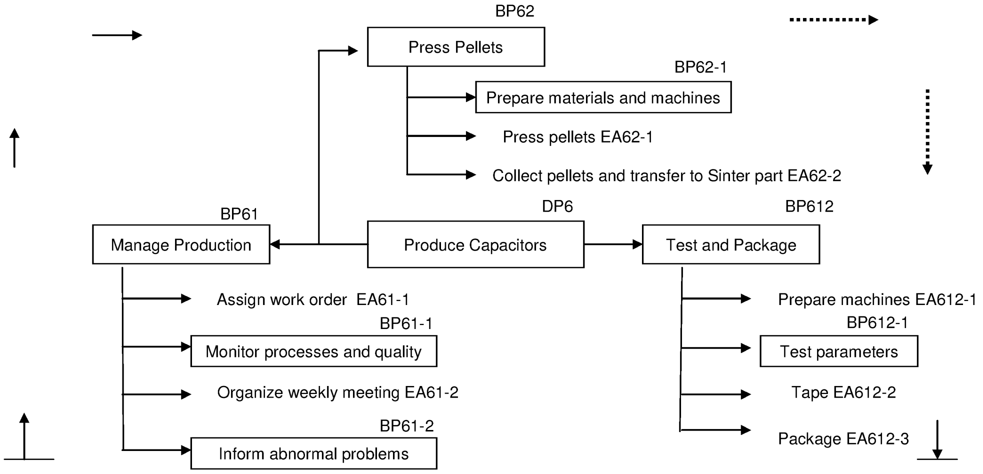


Figure 3 A part of Structure diagram for "Produce Capacitors" Domain Process

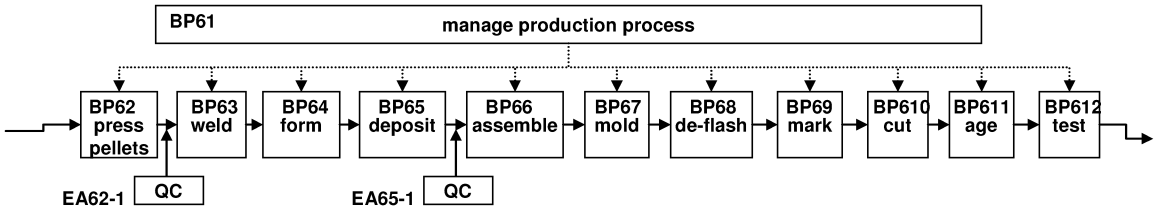


Figure 4 (a) Overview activity diagram of the CAP production process

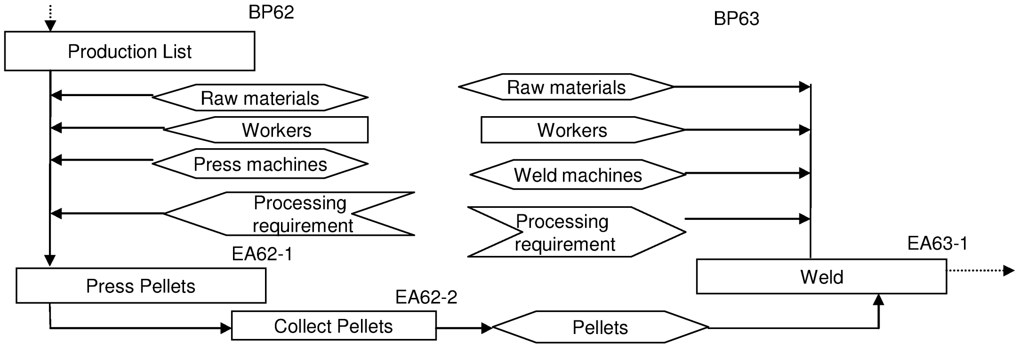


Figure 4(b) Expanded activity diagram showing the link between BP62 and BP63

CURRENT PROBLEMS FACING CAP

Though relative to many of its competitors CAP is a well organised and Hi-Tech Enterprise (with ISO9001, ISO14001 and OSHMS28001 certification), and has used management software in support of sales, supply, stock replenishment and financial processes, it still needs further improvement to:

1. minimise production bottle-necks and achieve its designed output capacity;
2. invariably meet customer order due dates to maximise profit and reputation;
3. respond more rapidly to uncertainty in its environment, by planning and implementing organisational change more effectively and quickly;
4. be well placed to benefit from the rapid and effective adoption of new generation production management software with minimum investment cost and maximum performance improvements.

From the above high level problem descriptions, it was observed that CAP needs to (1) use suitable tools to identify in advance real problem causes and avoidance strategies, (2) more specifically to identify improved scheduling and dispatching methods that closely match the organisation's

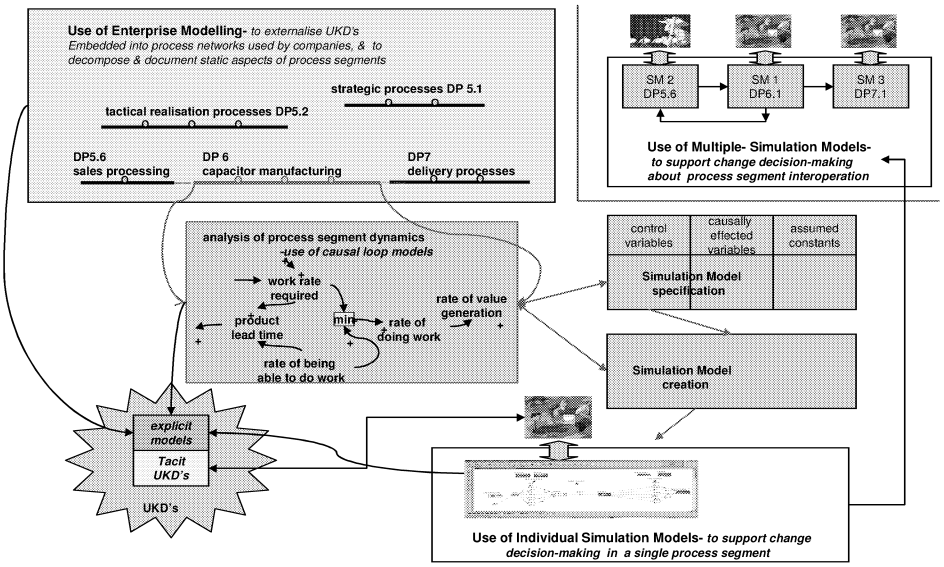

Use of Enterprise Modelling- to externalise UKD's Embedded into process networks used by companies, & to decompose & document static aspects of process segments
strategic processes DP 5.1
tactical realisation processes DP5.2
DP5.6 sales processing
DP 6 capacitor manufacturing
DP7 delivery processes
SM 2 DP5.6
SM 1 DP6.1
SM 3 DP7.1
Use of Multiple- Simulation Models- to support change decision-making about process segment interoperation
analysis of process segment dynamics -use of causal loop models
work rate required
product lead time
min
rate of doing work
rate of value generation
rate of being able to do work
control variables
causally effected variables
assumed constants
Simulation Model specification
Simulation Model creation
explicit models
Tacit UKD's
UKD's
Use of Individual Simulation Models- to support change decision-making in a single process segment

Figure 5 Systematic Capture & Deployment of Collective Organisational Understandings, Knowledge and Data

needs as they change over time (Heizer and Render 1990). Therefore the case study reported in this paper was conceived with the objective of prototyping the use of a well structured modelling approach, centred on simulation modelling: to solve CAP problems related to (1) and (2) on an ongoing basis. Figure 5 illustrates the modelling methodology adopted (Chatha and Weston 2005, Rahimifard and Weston 2005) : firstly Enterprise Modelling (EM) is used to document a 'big picture' about CAP; then modelling is focused on one or more segments that CAP urgently needs to improve, by using causal loop models to analyse the factors which affect the performance of this part; also through combined use of CIMOSA and causal loop models to gain a consensus view of real problem causes and potential avoidance strategies; and finally to develop and deploy focussed simulation models to test the avoidance strategies, such as by predicting outcomes from new resource arrangements and/or from adopting new scheduling systems.

SCOPE AND FOCUS OF SIMULATION MODELLING

Although the EM of CAP provided an explicit description of relatively enduring (i.e. static aspects) of its current production processes (and their structural links to other CAP processes) it was observed not to encode time dependency aspects of CAP entities and entity relationships. Therefore causal loop modelling was used to (a) explicitly document complementary temporal and causal relationships within CAP and (b) facilitate problem understandings and avoidance strategy development. The starting point here was that of using what was believed to be semi generic causal loop models, previously developed as an abstraction of cause and effect relationships observed within processes used by a furniture making SME.

Figure 6 shows basic causal relationships that are considered to impact on dynamic behaviours of CAP's production processes, which currently run primarily on a make-to-order basis. This figure identified dependencies between key process variables that needed to be characterised and parameterised by studying real executing production process segments, resource systems used and work pattern input and output behaviours. For example in the case of the 'rate of doing work' process variable, work

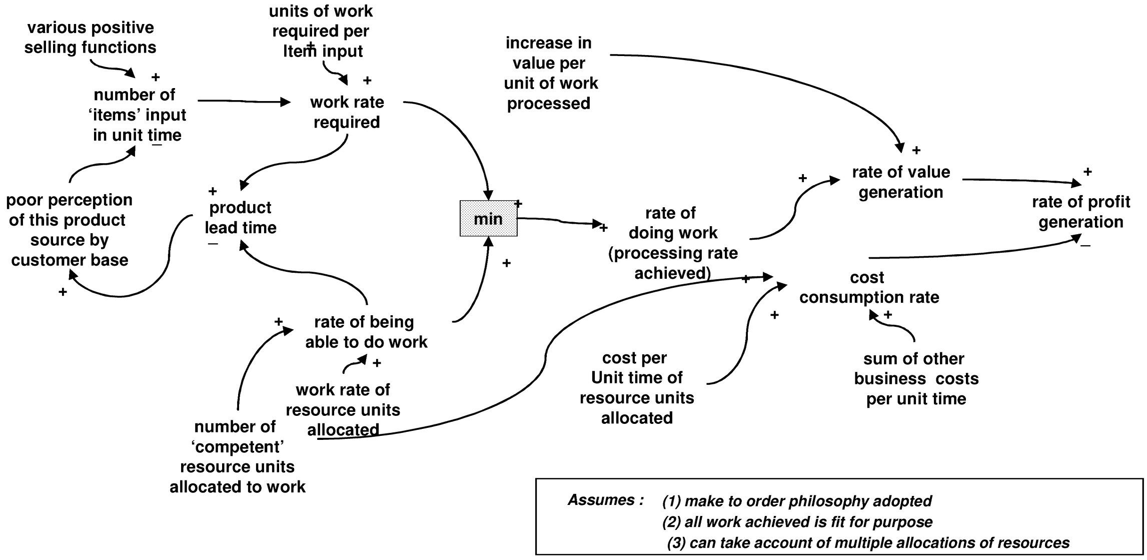

various positive selling functions
number of 'items' input in unit time
units of work required per Item input
work rate required
increase in value per unit of work processed
rate of value generation
rate of profit generation
poor perception of this product source by customer base
product lead time
min
rate of doing work (processing rate achieved)
cost consumption rate
rate of being able to do work
work rate of resource units allocated
number of 'competent' resource units allocated to work
cost per Unit time of resource units allocated
sum of other business costs per unit time
Assumes : (1) make to order philosophy adopted
(2) all work achieved is fit for purpose
(3) can take account of multiple allocations of resources

Figure 6 Basic Causal Relationships Impacting on the Production Processes Used by Cap

study results were needed to establish causal impacts on 'operation times' on the type and number of resource units allocated. Also because of the study emphasis on planning and scheduling production processes, Figure 7 was developed to begin to characterise and parameterise causal effects of other process variables on the 'rate of being able to do work' process variable.

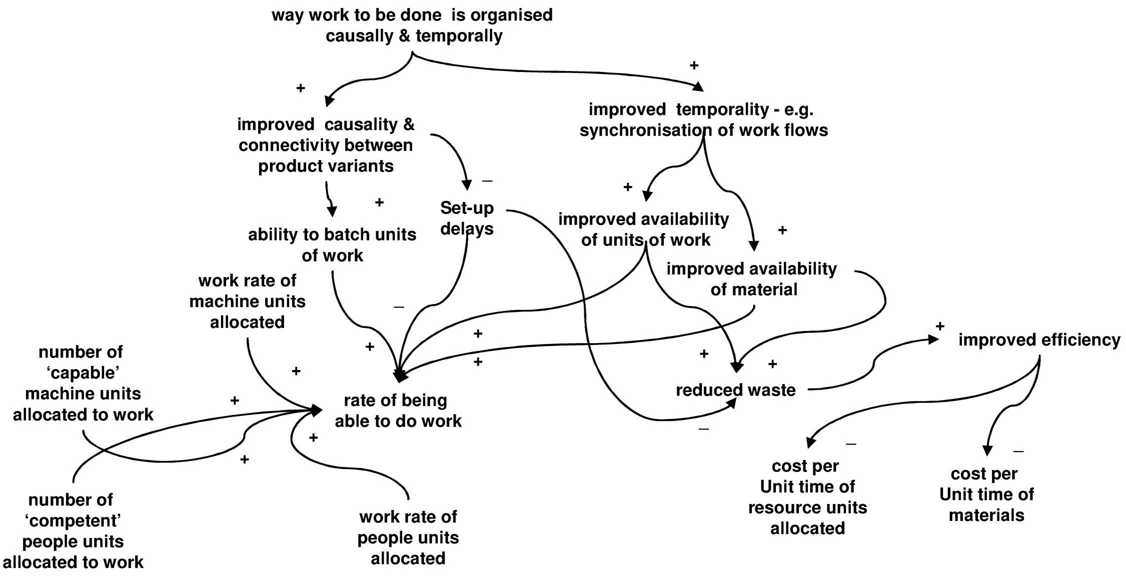

way work to be done is organised causally & temporally
improved causality & connectivity between product variants
improved temporality - e.g. synchronisation of work flows
ability to batch units of work
Set-up delays
improved availability of units of work
improved availability of material
work rate of machine units allocated
number of 'capable' machine units allocated to work
rate of being able to do work
reduced waste
improved efficiency
number of 'competent' people units allocated to work
work rate of people units allocated
cost per Unit time of resource units allocated
cost per Unit time of materials
+
–

Figure 7 Basic Causal Relationships: impacting on the 'rate of being able to do work' at CAP

Having used causal loops to understand dynamic properties of the production processes used by CAP, the authors could now design and implement:

(i) a general purpose simulation model of CAP's make to order production processes

(ii) a set of simulation modelling experiments to (a) validate behaviours of the general purpose simulation model under specific operating conditions and (b) predict new behaviours should alternative scheduling methods be adopted.

GENERAL PURPOSE SIMULATION MODEL DEVELOPMENT: TO MODEL BEHAVIOURS OF CAP'S PRODUCTION PROCESSES

It follows that previous case study modelling steps had created and validated (1) an EM providing a big picture of all relatively enduring (i.e. 'static') aspects of (day to day) operational processes currently used in CAP, including its make to order capacitor production processes and (2) causal loops related to dynamic aspects of capacitor production processes in CAP, which characterise in qualitative terms likely effects of changing production process variables. The EM visually identified many sequences of activities that need to be resourced within CAP by competent people and capable machines. Naturally also the EM templates were observed to provide means of 'attaching' (a) visual models of information, material and control flows and (b) tabulated lists (e.g. within spread sheets) of human and technical resource requirements and assignments, and related operation times and set up times, when the activity sequences are performed to realise different capacitor types and variants within types. Whereas the causal loop models were found to usefully encode understanding about likely impacts of applying different time dependent (machine and people) resourcing policies and alternative ways of organising workflows through segments and elemental activities of production processes. Also having determined foci of study (such as to consider the impact of different scheduling and resourcing policies) the causal loops help to identify those key causal and temporal relationships that needed to be understood in greater detail so that relevant parametric data could be obtained before any form of quantitative analysis could be attempted.

Based on the foregoing analysis a general purpose simulation model of make to order production processes at CAP was built using a proprietary discrete event simulator (namely Simul8). This simulation model is general purpose in that it facilitates experimentation (with respect to needed process variables identified during causal loop modelling), enabling impact analysis of changes to scheduling and work organisation policies. A set of simulation modelling experiments were designed and carried out so as to test effects of policy change on 'inventory levels', 'bottlenecks', 'resource utilisation', 'value generation' and 'process costs'(Bozzone 2001).

Figure 8 shows the main 'building blocks' of the simulation model constructed for the above purpose. These 'blocks' have a one-to-one correspondence with activities graphically illustrated by Figure 5; these that relate BP62 and BP63 process segments. The 'building blocks' were realised using standard Simul8 modelling constructs; namely 'work centre', 'process route', 'work entry point', 'work exit point' and 'queue' model constructs. Each 'building block' was modelled in a similar 'modular' fashion, to facilitate future change and reuse of process models & process parameters.

One key design issue when creating the general purpose model was related to deciding how to consolidate different product types and variants, into product groups that need to be modelled uniquely; such that sufficient confidence in

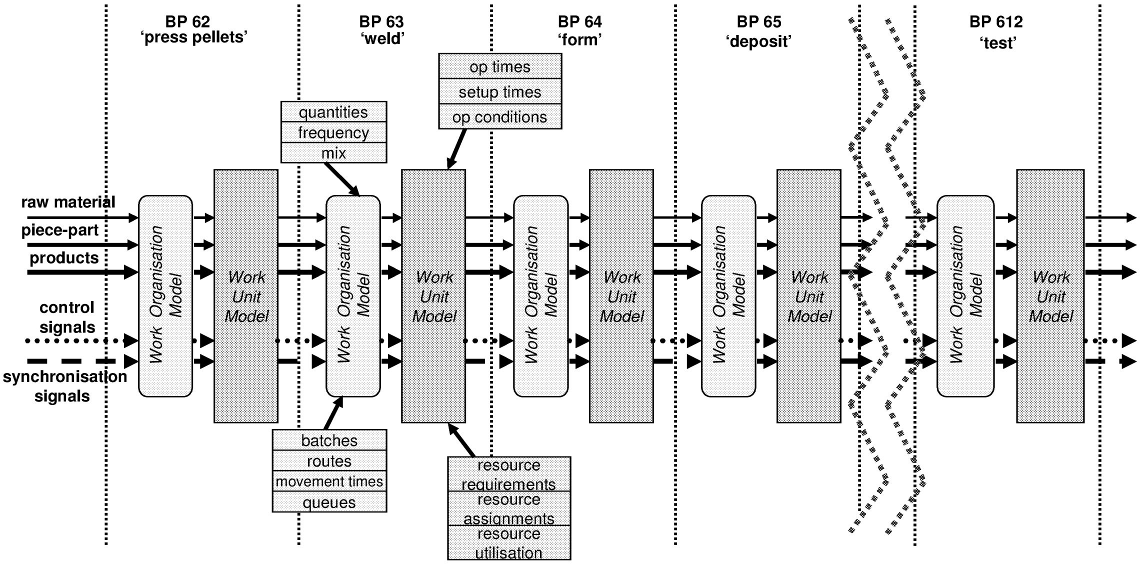


Figure 8 Modular Building Blocks of the CAP Production Process Simulation Model

modelled results could be obtained. Decisions made here were resolved by running the general purpose model whilst replicating current resourcing and scheduling policies used by CAP, inputting various historical work patterns (with alternative product groupings) to modelled work entry points, and observing differences between modelled production behaviours and actual historical production behaviours realised by CAP.

Experimental use of the general purpose simulation model of CAPs make to order production processes is ongoing, and is centred on predicting both production and business outcomes from using different policies which include: push/forward, pull/backward and postponement/hybrid approaches.

When designing experiments to investigate the impacts of these policy changes the causal loop models help distinguish between 'control variables', 'causally impacted variables' and 'constants' in the production process modelled. Here 'control variables' are required to realise the policies under study, while 'causally impacted variables' naturally occur as a consequence of control variable change. Table 1 lists control and causally impacted variables identified for the policies currently under study.

Table 1

	Forward	Backward	Postponement
Control Variables	Resource Assignments Work rate	Customer Order Mix Resource Assignments Work rate	Batch-size Customer Order Mix Resource Assignments Work rate
Main Impacted Variables	Lead-time Cost	Outputs Quantity Cost	Due-time Lead-time Cost

GENERAL OBSERVATIONS

This paper illustrates complexities involved when developing and using simulation models to inform aspects of organisation design and change. Also illustrated is a structured approach to modelling which results in 'well situated' simulation models; where the context of the model, the model's purpose and scope and many of its parameters and parameter relationships are well defined. In principle this should improve the reusability of such models and the quality of predictions made; as they can be interpreted from alternative organisation-wide perspectives. Whereas currently in manufacturing companies current best practice is to use simulation models in a piecemeal, ad hoc fashion, which limits benefits obtained and a wider industrial use of simulation techniques.

This paper also shows how simulation modelling complements other techniques such that organisational understanding, knowledge and data can become more accessible and reusable; leading to significant competitive advantage. The downside is significant cost (of people time and expertise) when eliciting and organising data into needed model forms; and later when updating models as organisational changes occur. In theory the ideas, methods and models introduced herein can provide a useful step towards better and faster organisation design and change

REFERENCES

AMICE Consortium, 1993, *CIMOSA: Open System Architecture for CIM,* 2nd Ed Springer-Verlag, Berlin.

J Heizer & B Render 1990 Production and Operations Management:Strategies and Tactics, Second Edition

K.A.Chatha & R.H.Weston 2005 'Combined Enterprise and Simulation Modelling in Support of Process Engineering' IJCIM 18-8.

A Rahimifard & R.H.Weston 2005 'Enhanced Use of Enterprise and Simulation Modelling Techniques to Support Factory Changeability', CIRP 05 Conf.

V Bozzone, 2001 'Speed to market: lean manufacturing for job shops' 2nd Ed.

COMPLEX SYSTEMS MODELLING AND METHODOLOGY

Different goals in multiscale simulations and how to reach them

Pierrick Tranouez and Antoine Dutot
LITIS
Université du Havre
UFR Sciences et Techniques
25 rue Ph. Lebon - BP 540
76058 Le Havre Cedex – France
Pierrick.Tranouez@univ-lehavre.fr

ACKNOWLEDGEMENTS

Now they are grown-ups and write their articles alone, the authors would like to thank Cyrille Bertelle and Damien Olivier, former PhD supervisors, who could be credited as co-author for all they did on this subject. But they will not. One has to start at some time.

KEYWORDS

Multiscale, clustering, dynamic graphs, adaptation

ABSTRACT

In this paper we sum up our works on multiscale programs, mainly simulations. We first start with describing what multiscaling is about, how it helps perceiving signal from a background noise in a flow of data for example, for a direct perception by a user or for a further use by another program. We then give three examples of multiscale techniques we used in the past, maintaining a summary, using an environmental marker introducing an history in the data and finally using a knowledge on the behavior of the different scales to really handle them at the same time.

INTRODUCTION: WHAT THIS PAPER IS ABOUT, AND WHAT IT'S NOT

Although we delved into different applications and application domains, the computer science research goals of our team has remained centered on the same subject for years. It can be expressed in different ways that we feel are, if not exactly equivalent, at least closely connected. It can be defined as managing multiple scales in a simulation. It also consists in handling emergent structures in a simulation. It can often also be seen as dynamic heuristic clustering of dynamic data[1]. This paper is about this theme, about why we think it is of interest and what we've done so far in this direction. It is therefore akin to a state of the art kind of article, except more centered on what we did. We will allude to what others have done, but the focus of the article is presenting our techniques and what we're trying to do, like most articles do, and not present an objective description of the whole field, as the different applications examples could make think : we're sticking to the same computer science principles overall. We're taking one step back from our works to contemplate them all, and not the three steps which would be necessary to encompass the whole domain, as it would take us beyond the scope of the conference.

PERCEPTION: FILTERING TO MAKE DECISIONS

I look at a fluid flow simulation but all I'm interested in is where does the turbulence happen, in a case where I couldn't know before the simulation (Tranouez et al. 2005). I use a multi-participant communication system in a crisis management piece of software and I would like to know what are the main interests of each communicant based on what they are saying (Lesage et al. 1999). I use an IBM model of different fish species but I'm interested in the evolution of the populations, not the individual fish (Prevost et al. 2004). I use a traffic simulation with thousands of cars and a detailed town but what I want to know is where the traffic jams are (coming soon).

In all those examples, I use a piece of software which produces huge amounts of data but I'm interested in phenomena of a different scale than the raw basic components. What we aim at is helping the user of the program to reach what he is interested in, be this user a human (Clarification of the representation) or another program (Automatic decision making). Although we're trying to stay general in this part, we focused on our past experience of what we actually managed to do, as described in "Some techniques to make these observations in a time scale comparable to the observed", this isn't gratuitous philosophy.

Clarification of the representation

This first step of our work intends to extract the patterns on the carpet from its threads (Tranouez 1984). Furthermore, we want it to be done in "real (program) time", meaning not a posteriori once the program is ended by examining its traces (Servat et al; 1998), and sticking as close as possible to the under layer, the one pumping out dynamic basic data. We don't want the discovery of our structures to be of a greater time scale than a step of the program it works upon.

How to detect these structures? For each problem the structure must be analyzed, to understand what makes it stand out for the observer. This implies knowing the observer purpose, so as to characterize the structure. The answers are problem specific, nevertheless rules seem to appear.

In many situations, the structures are groups of more basic entities, which then leads to try to fathom what makes it a

[1] We will of course later on describe in more details what we mean by all this.

group, what is its inside, its outside, its frontier, and what makes them so.

Quite often in the situation we dealt with, the groups members share some common characteristics. The problem in that case belongs to a subgenre of clustering, where the data changes all the time and the clusters *evolve* with them, they are not computed from scratch at each change.

The other structures we managed to isolate are groups of strongly communicating entities in object-oriented programs like multiagent simulations. We then endeavored to manage these cliques.

In both cases, the detected structures are emphasized in the graphical representation of the program. This clarification lets the user of the simulation understand what happens in its midst. Because modeling, and therefore understanding, is clarifying and simplifying in a chosen direction a multi-sided problem or phenomenon, our change of representation participates to the understanding of the operator. It is therefore also a necessary part of automating the whole understanding, aiming for instance at computing an artificial decision making.

Automatic decision making

Just like the human user makes something of the emerging phenomena the course of the program made evident, other programs can use the detected organizations.

For example in the crisis management communication program, the detected favorite subject of interest of each of the communicant will be used as a filter for future incoming communications, favoring the ones on connected subjects. Other examples are developed below, but the point is once the structures are detected and clearly identified, the program can use models it may have of them to compute its future trajectory. It must be emphasized that at this point the structures can themselves combine into groups and structures of yet another scale, recursively. We're touching there an important component of complex system (Simon 1996). We may hope the applications of this principle to be numerous, such as robotics, where perceiving structures in vast amounts of data relatively to a goal, and then being able to act upon these accordingly is a necessity.

We're now going to develop these notions in examples coming from our past works.

SOME TECHNIQUES TO MAKE THESE OBSERVATIONS IN A TIME SCALE COMPARABLE TO THE OBSERVED

The examples of handling dynamic organization we chose are taken from two main applications, one of a simulation of a fluid flow, the other of the simulation of a huge cluster of computed processes, distributed over a dynamic network of computing resources, such as computers. The methods titled "Maintaining a summary of a simulation" and "Reification: behavioral methods" are theories from the hydromechanics simulation, while "Traces of the past help understand the present" refers to the computing resources management simulation. We will first describe these two applications, so that an eventual misunderstanding of what they are doesn't hinder later the clarity of our real purpose, the analysis of multiscale handling methods.

In a part of a more general estuarine ecosystem simulation, we developed a simulation of a fluid flow. This flow uses a particle model (Leonard 1980), and is described in details in (Tranouez 2005) or (Tranouez et al. 2005). The basic idea is that each particle is a vorticity carrier, each interacting with all the others following Biot-Savart laws. As fluid flows tend to organize themselves in vortices, from all spatial scales from a tens of angstrom to the Atlantic Ocean, this is these vortices we tried to handle as the multiscale characteristic of our simulation. The two methods we used are described below.

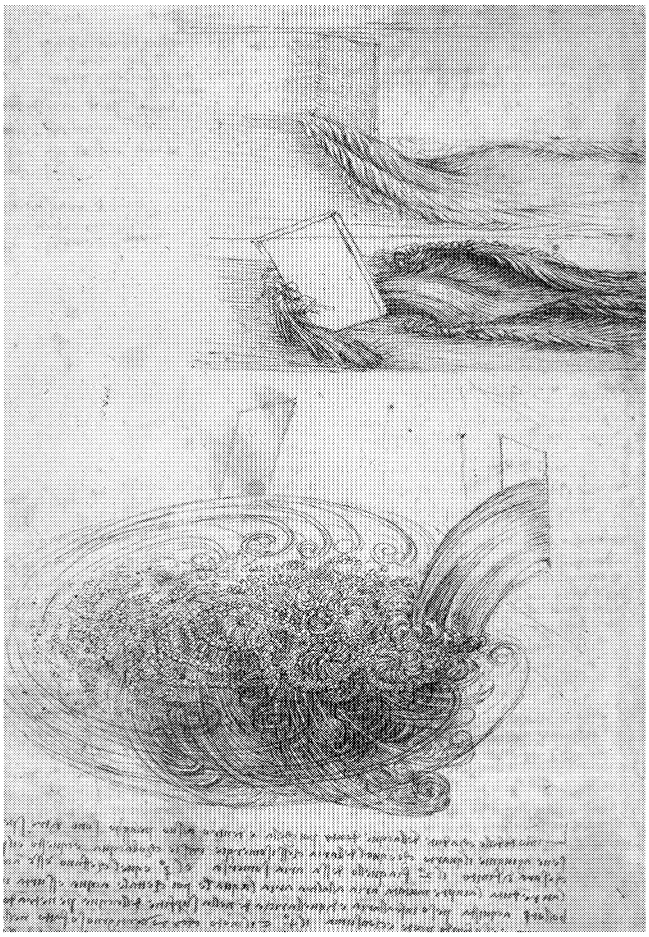


Figure 1: Vortices in a fluid flow by Leonardo Da Vinci

The other application, described in depth in (Dutot 2005), is a step toward automatic distribution of computing over computing resources in difficult conditions, as:

- The resources we want to use can each appear and disappear, increase or decrease in number.
- The computing distributed is composed of different object-oriented entities, each probably a thread or a process, like in a multiagent system for example (the system was originally imagined for the ecosystem simulation alluded to above, and the

entities would have been fish, plants, fluid vortices etc., each acting, moving …)

Furthermore, we want the distribution to follow two guidelines:

- As much of the resources as possible must be used,
- Communications between the resources must be kept as low as possible, as it should be wished for example if the resources are computers and the communications therefore happen over a network, bandwidth limited if compared to the internal of a computer.

This the ultimate goal of this application, but the step we're interested in today consists in a simulation of our communicating processes, and of a program which, at the same time the simulated entities act and communicate, advises how they should be regrouped and to which computing resource they should be allocated, so as to satisfy the two guidelines above.

Maintaining a summary of a simulation

The first method we would like to describe here relates to the fluid flow simulation. The hydrodynamic model we use is based on an important number of interacting particles. Each of these influences all the others, which makes n^2 interactions, where n is the number of particles used. This makes a great number of computations. Luckily, the intensity of the influence is inversely proportional to the square of the distance separating two particles. We therefore use an approximation called fast multipoles method, which consists in covering the simulation space with grids, of a density proportional to the density of particles (see Figure 2). The computation of the influence of its colleagues over a given particle is then done exactly for the ones close enough, and averaged on the grid for those further. All this is absolutely monoscale.

As the particles are vorticity carriers, it means that the more numerous they are in a region of space, the more agitated the fluid they represent is. We would therefore be interested in the structures built of close, dense particles, surrounded by sparser ones. A side effect of the grids of the FMP, is that they help us do just that. It's not that this clustering is much easier on the grids, it's above all that they are an order of magnitude less numerous, and organized in a tree, which makes the group detection much faster than if the algorithm was ran on the particles themselves. Furthermore, the step by step management of the grids is not only cheap (it changes the constant of the complexity of the particles movement method but not the order) but also needed for the FPM.

We therefore detect structures on

- Dynamic data (the particles characteristics)
- With little computing added to the simulation,

which is what we aimed at.

The principle here is that through the grids we maintain a summary of the simulation, upon which we can then run static data algorithm, all this at a cheap computing price.

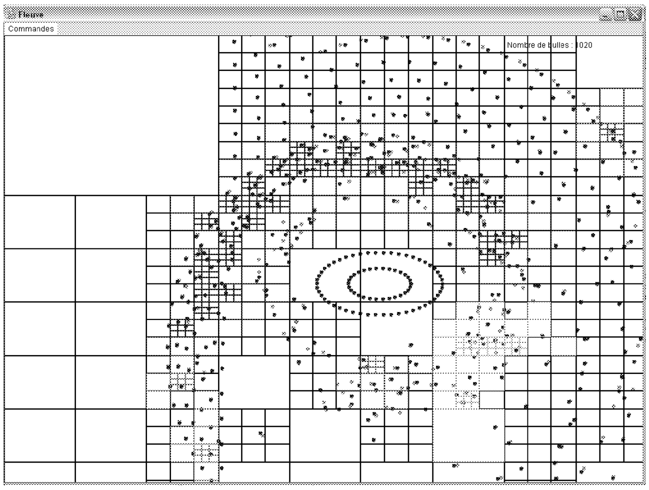


Figure 2 : Each color corresponds to a detected aggregate

Traces of the past help understand the present

The second method relates to the detection of communication clusters inside a distributed application. The applications we are interested in are composed of a large number of object-oriented entities that execute in parallel, appear, evolve and, sometimes, disappear. Aside some very regular applications, often entities tend to communicate more with some than with others. For example in a simulation of an aquatic ecosystem, entities representing a species of fish may stay together, interacting with one another, but flee predators. Indeed organizations appear groups of entities form. Such simulations are a good example of applications we intend to handle, where the number of entities is often too large to compute a result in an acceptable time on one unique computer.

To distribute these applications it would be interesting to both have approximately the same number of entities on each computing resource to balance the load, but also to avoid as much as possible to use the network, that costs significantly more in terms of latency than the internals of a computer. Our goal is therefore to balance the load and minimize network communications. Sadly, these criteria are conflicting, and we must find a tradeoff.

Our method consists in the use of an ant metaphor. Applications we use are easily seen as a graph, which is a set of connected entities. We can map entities to vertices of the graph, and communications between these entities to the edges of the graph. This graph will follow the evolution of the simulation. When an entity appear, a vertex will appear in the graph, when a communication will be established between two entities, an edge will appear between the two corresponding vertices. We will use such a graph to represent the application, and will try to find clusters of highly communicating entities in this graph by coloring it, assigning a color to each cluster. This will allow to identify clusters as a whole and use this information to assign not entities, but at another scale, clusters to computing resources.

For this, we use numerical ants that crawl the graph as well as their pheromones, olfactory messages they drop, to mark clusters of entities. We use several distinct colonies of ants, each of a distinct color, that drop colored pheromones. Each color corresponds to one of the computing resources at our disposal. Ants drop colored pheromones on edges of the graph when they cross them. We mark a vertex as being of the color of the dominant pheromone on each of its incident edges. The color indicates the computing resource where the entity should run.

To ensure our ants color groups of highly communicating entities of the same color to minimize communications, we use the collaboration between ants: ants are attracted by pheromones of their own color, and attracted by highly communicating edges. To ensure the load is balanced, that is to ensure that the whole graph is not colored only in one color if ten colors are available, we use competition, ants are repulsed by the pheromones of other colors.

Pheromones in nature being olfactory molecules, they tend to evaporate. Ants must maintain them so they do not disappear. Consequently, only the interesting areas, zones where ants are attracted, are covered by pheromones and maintained. When a zone becomes less interesting, ants leave it and pheromone disappear. When an area becomes of a great interest, ants colonize it by laying down pheromones that attract more ants, and the process self-amplifies.

We respect the metaphor here since it brings us the very interesting property of handling the dynamics. Indeed, our application continuously changes, the graph that represents it follows this, and we want our method to be able to discover new highly communicating clusters, while abandoning vertices that are no more part of a cluster. As ants continuously crawl through the graph, they maintain the pheromone color on the highly communicating clusters. If entities and communications of the simulation appear or disappear, ants can quickly adapt to the changes. Colored pheromones on parts where a cluster disappeared evaporate and ants colonize new clusters in a dynamic way. Indeed, the application never changes completely all the time; it modifies itself smoothly. Ants lay down "traces" of pheromones and do not recompute the color of each vertex at each time, they reuse the already dropped pheromone therefore continuously giving a distribution advice at a small computing price, and adapting to the reconfigurations of the underlying application.

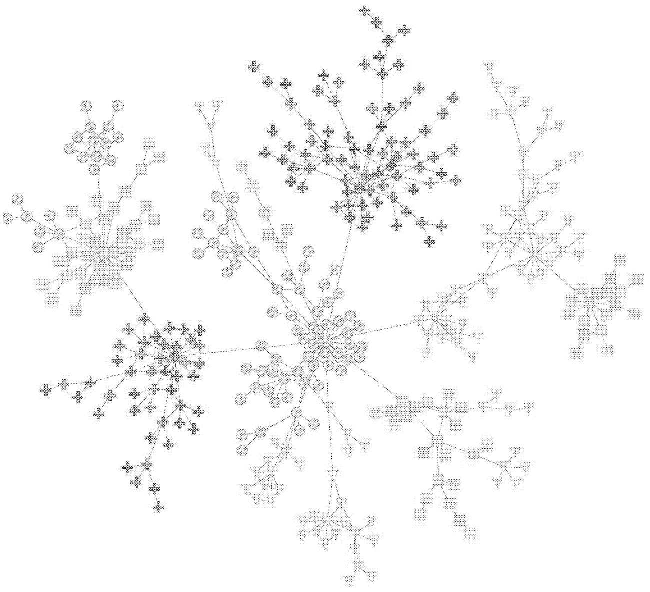


Figure 3: Each color corresponds to a computing ressource

Reification: behavioral methods

This last example of our multiscale handling methods was also developed on the fluid flow simulation. Once more, we want to detect structures in a dynamic flow of data, without getting rid of the dynamicity by doing a full computation on each step of the simulation. The idea here is doing the full computation only once in a good while, and only relatively to the unknown parts of our simulation.

We begin with detecting vortices on the basic particles once. Vortices will be a rather elliptic set of close particles of the same rotation sense. We then introduce a multiagent system of the vortices. We have indeed a general knowledge of the way vortices behave. We know they move like a big particle in our Biot-Savard model, and we model its structural stability through social interactions with the surrounding basic particles, the other vortices and the obstacles, through which they can grow, shrink or die (be dissipated into particles). The details on this can be found in (Tranouez et al. 2005). Later on, we occasionally make a full-blown vortex detection, but only on the remaining basic particles, as the already detected vortexes are managed by the multiagent system

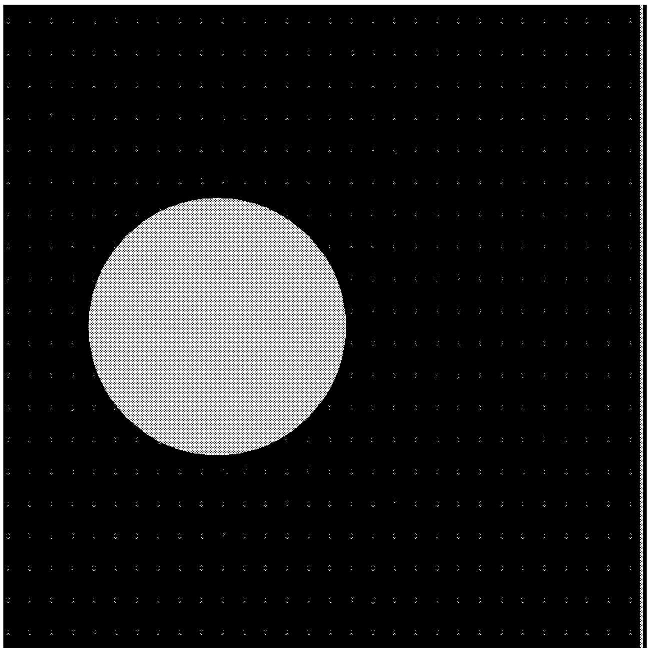


Figure 4: Fluid flow around an obstacle, initial state

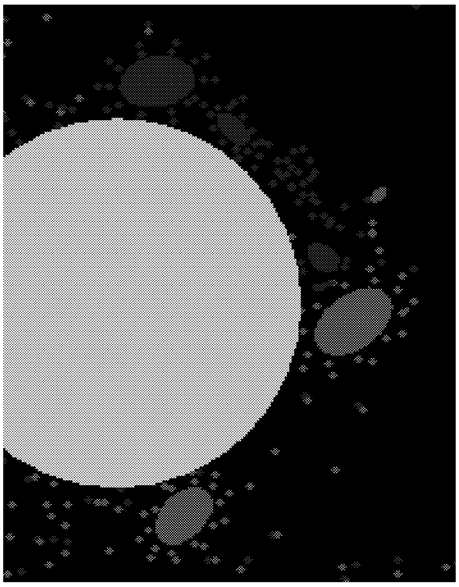


Figure 5: Part of the flow, some steps later. The ellipses are vortices.

In this case, we possess knowledge on the structures we want to detect, and we use it to build actually the upper scale level of the simulation, which at the same time lightens ulterior structures detection. We are definitely in the category described in Automatic decision making.

CONCLUSION

Our research group works on complex systems and focuses n the computer representation of their hierarchical/holarchical characteristics (Koestler 1978) (Simon 1996) (Kay 2000). We tried to illustrate that describing a problem at different scales is a well-spread practice at least in the modeling and simulating community. We then presented some methods for handling the different scales, with maintaining a summary, using an environmental marker introducing a history in the data and finally using knowledge on the behavior of the different scales to handle them at the same time.

We now believe we start to have sound multiscale methods, and must focus on the realism of the applications, to compare the sacrifice in details we make when we model the upper levels rather than just heavily computing the lower ones. We save time and lose precision, but what is this trade-off worth *precisely*?

REFERENCES

Dutot A., *Distribution dynamique adaptative à l'aide de mécanismes d'intelligence collective*, PhD thesis, Le Havre University

Kay J., « Ecosystems as Self-Organising Holarchic Open Systems : narratives and the second law of thermodynamics », S.E.JORGENSEN, MÜLLER F., Eds., *Handbook of Ecosystems Theories and Management*, Lewis Publishers, 2000.

Koestler A. 1978, *Janus. A Summing Up* 1978

Leonard A. 1980, « Vortex methods for flow simulation », *Journal of Computational Physics*, vol. 37, 1980, p. 289-335.

Lesage F., Cardon A., Tranouez P. : "A multiagent based prediction of the evolution of knowledge with multiple points of view"; KAW' 99; (1999)

Prevost G., Tranouez P., Lerebourg S., Bertelle C., Olivier D. 2004 : "Methodology for holarchic ecosystem model based on ontological tool"; ESMC 2004; pp 164-171 (2004)

Servat D., Perrier E., Treuil J.-P., Drogoul A. 1998, « When Agents Emerge from Agents: Introducing Multi-scale Viewpoints in Multi-agent Simulations », *MABS*, 1998, p. 183-198.

Simon H. 1996, *The Sciences of the Artificial (3rd Edition)* MIT Press

Tranouez Pierre 1984, *Fascination et narration dans l'œuvre romanesque de Barbey d'Aurevilly* , Doctorat d'État

Tranouez Pierrick 2005, *Penicillo haere, nam scalas aufero*, PhD thesis, Le Havre University

Tranouez P., Bertelle C. and Olivier D. 2005, « Changing levels of description in a fluid flow simulation", EPNADS 2005

OPTIMIZATION IN PACKAGING AND REAL ESTATE

William C. Conley
Business Administration
University of Wisconsin at Green Bay
Green Bay, Wisconsin 54311
U.S.A.
E-mail: conleyw@uwgb.edu

KEYWORDS
Multi stage optimization, packaging, real estate

ABSTRACT

Multi stage Monte Carlo optimization (MSMCO) is a general purpose simulation optimization approach that is used on complex multivariate problems. It is a several stage simulation approach that finds and stores a best solution so far at each stage of the simulation. Each subsequent stage is centered about the previous best answer produced so far. The search continues in a reduced region about this best answer found so far. The subsequent stages keep finding better and better answers as the simulation proceeds. Presented here as examples will be a packaging optimization problem followed by a real estate problem involving 100 acres of land. Finally, a third example will use MSMCO on a statistical analysis of four economic indicators that may be correlated with sales of real estate.

INTRODUCTION

Desktop computers have become so powerful and inexpensive in the twenty-first century that it is now possible to run massive simulations involving millions or billions of sample solutions in a few minutes of computer time. This makes a self-improving genetic algorithm type simulation like multi stage Monte Carlo optimization a viable technique for a wide variety of business, science and engineering optimization problems. Presented here will be a packaging optimization problem involving containerized shipping. Next, a real estate optimization problem and a statistical analysis of factors that may contribute or be correlated to increased real estate sales. Multi stage Monte Carlo optimization (MSMCO) is the simulation technique used on all three of these problems.

A PACKAGING EXAMPLE

A packaging company wants to design ten different size rectangular box containers that are to be stacked in a large containerized shipping box of 63,000,000 cubic centimeters volume. The dimensions are 3 x 3 x 7 meters. However, because some of the ten different size boxes may have to be loaded inside the 3 x 3 x 7 = 63 cubic meter container (for shipping) with some of the other size boxes, it is desired that the stacking of them be easier for the workers. Therefore, management wants at least two dimensions of each type of box to match two dimensions of one of the other size boxes. Also, the volume of the ten boxes should be about one quarter of the volume of the large container. Additionally, it is desired that several of each type of box be able to fit inside the container together and that all box dimensions be between 10 and 500 centimeters. Therefore to get started, managers attempt to solve the system of Equations (1) and (2) below.

$$\sum_{i=1}^{10} X_i X_{i+1} X_{i+2} = 15{,}750{,}000 \qquad (1)$$

$$\sum_{i=1}^{10} iX_i X_{i+1} X_{i+2} = 63{,}000{,}000 \qquad (2)$$

subject to

$$10 \le X_i \le 500 \text{ for } I = 1, 2, 3\ldots 10$$

The system is transformed to minimize the sum of the absolute values of the differences between the left and right hand side of each equation. A one page multi stage Monte Carlo simulation drawing 25,000 feasible solutions (in an ever narrowing and repositioning search region) funnels into several approximate solutions in a few minutes of computer time.

Four possible solutions to this ten box problem are given below in the pattern of:

X_1	X_2	X_3	X_4
X_5	X_6	X_7	X_8
X_9	X_{10}	X_{11}	X_{12}
equation one error	equation two error		

454.502	46.482	388.007	14.260
111.140	75.468	186.506	40.799
103.820	31.736	269.984	304.461
$e_1 = .0358$	$e_2 = .0339$		
163.152	131.512	160.928	222.024
66.895	34.753	102.359	65.308
59.383	135.564	204.041	59.878
$e_1 = .0268$	$e_2 = .0231$		
285.349	339.595	39.157	251.453
50.717	196.807	129.396	51.787
70.053	109.227	145.315	63.927
$e_1 = .0225$	$e_2 = .0001$		
52.793	308.066	55.734	476.360
29.779	87.540	285.652	18.379
49.806	117.429	233.313	61.846
$e_1 = .0103$	$e_2 = .0205$		

This solution approach is assuming that the 3 x 3 x 7 = 63 cubic meters large shipping container has inside dimensions of 3, 3 and 7 meters. However, the X_1, X_2, $X_3 \ldots X_{10}$ solutions for the little boxes are assumed to be outside dimensions. If the thickness of the cardboard or wood or metal used to make the containers is an issue, that additional thickness could be added to the variables in the equations.

Just to illustrate our printout a bit, 454.502 x 46.482 x 388.007 would be one of the box size dimensions. Another one would be 46.482 x 388.007 x 14.260. A third one would be 14.260 x 111.140 x 75.468 centimeters and so forth. This way the boxes would have many common dimensions for ease of stacking together.

A REAL ESTATE PROBLEM

A real estate developer has a one hundred acre parcel of land in the shape of a square of length 2087 feet on each side. He desires to divide it up into dozens of different lots of nine different lot sizes such that each type of lot has at least one rectangular dimension the same as one of the eight other lot sizes for ease of planning. Also, the sum of the nine lot sizes is to equal 400,000 square feet for planning purposes. Additionally, many different numbers of each lot size should add up to the one hundred acres or 2087 x 2087 square feet. Also, there are to be at least five of each type of lot. Therefore, the realtor uses multi stage Monte Carlo (MSMCO) to solve Equations (3) and (4).

$$\sum_{i=1}^{9} X_i X_{i+1} = 400{,}000 \qquad (3)$$

$$\sum_{i=1}^{9} (i+5) X_i X_{i+1} = 2087^2 \qquad (4)$$

and

$$50 \le X_i \le 400 \text{ feet}$$

as each lot dimension should be at least 50 feet and no longer than 400 feet. This will be a retirement community. There will be no roads, only golf cart paths and sidewalks and trails. Therefore, for planning purposes, their dimensions are considered negligible for now.

Four solutions in the pattern of:

X_1	X_2	X_3	X_4	X_5
X_6	X_7	X_8	X_9	X_{10}
equation one error	equation two error			

are

179.542	107.051	283.237	195.146	121.972
233.544	348.367	68.778	317.498	364.545
e_1=.0576	e_2=.0925			
102.521	95.042	103.799	321.072	222.265
183.031	314.826	244.759	208.123	237.362
e_1=.0285	e_2=.1462			
200.929	141.669	290.158	96.474	384.518
116.866	115.355	333.474	235.095	383.057
e_1=.0152	e_2=.0428			
84.99	234.341	86.879	280.248	305.254
177.746	172.572	89.729	320.294	376.757
e_1=.0873	e_2=.1190			

Therefore, looking at the upper left part of the printout, one lot is to have rectangular dimensions of 179.542 x 107.051 feet. Another one is to have dimensions of 107.051 x 283.237 feet. Another one could have dimensions of 283.237 x 195.146 feet, etc.

These real estate lot size solutions were produced with a fifteen stage MSMCO simulation drawing 20,000 feasible solutions at each stage. The four solutions of nine different lot sizes reported here are the stage 15 printout of four MSMCO runs of the program.

A SECOND REAL ESTATE EXAMPLE

A realtor wants to test her theory that real estate sales are correlated to (and are therefore perhaps a function of) four economic indicators. Table 1 shows the past 49 months of data on these variables. Columns one through four represent the four key economic indicator readings (on a scale of 0 to 100). These are variables one, two, three and four. Column five (representing variable five) is the average number of properties sold that month in each of the realtor's company offices nationwide. Therefore, she tests the hypothesis:

H_0: no correlation between
variables X_1, X_2, X_3, X_4 and X_5

H_A: a correlation between the variables

Therefore, the multi stage Monte Carlo optimization traveling salesperson (TSP) algorithm is used (adjusted for five dimensions and n = 49 lines of data) to find a shortest route connecting the n = 49 five dimensional points in a closed loop tour of total distance A = 1606.458. Please see Table 2 for the printout of the point to point route.

Table 1: The Monthly Data

	X_1	X_2	X_3	X_4	X_5
1	79	54	30	8	5.68
2	25	2	100	76	8.47
3	77	7	80	100	18.17
4	12	84	75	13	6.68
5	68	20	36	35	4.89
6	21	41	60	94	9.97
7	100	26	79	31	12.80
8	21	42	59	83	8.69
9	92	25	59	26	8.37
10	3	87	66	37	7.48
11	39	38	82	48	8.91
12	100	74	40	96	32.28
13	21	60	63	40	6.68
14	10	26	61	24	3.04
15	95	93	18	6	9.48
16	5	39	78	11	3.53
17	94	62	48	44	14.87
18	70	37	11	48	5.34
19	66	70	34	94	18.17
20	79	31	100	26	12.80
21	71	10	49	100	11.88
22	64	69	42	52	11.44
23	61	44	26	71	8.37
24	90	69	30	79	19.10
25	90	67	18	23	7.96
26	86	69	59	52	18.63
27	81	81	73	26	17.50
28	9	59	56	16	3.86
29	14	100	68	13	7.67
30	6	61	98	35	8.16
31	98	64	27	1	7.20
32	48	89	96	89	37.51
33	83	59	20	10	5.75
34	43	52	33	38	5.34
35	86	82	29	67	18.17
36	11	86	100	45	13.80
37	89	2	59	12	5.08
38	91	31	100	27	15.06
39	40	79	9	76	8.58
40	76	48	25	30	6.28
41	80	46	88	2	9.97
42	22	21	28	87	4.83
43	73	44	21	49	6.94
44	34	68	2	26	3.40
45	50	25	20	98	7.48
46	96	50	62	29	12.96
47	20	69	18	9	2.86
48	23	24	47	60	4.59
49	22	75	38	76	9.37

Table 2: The Shortest Route

A = 1606.458

2	3	21	45	42	6	8	48	14	16
28	47	44	34	5	37	9	7	38	20
41	46	17	26	27	15	31	1	33	25
40	43	18	23	22	35	24	12	19	39
49	32	36	30	29	4	10	13	11	2

Then four sets of n = 49 random data points (in the same ranges as the original data) are fed into the MSMCO traveling salesman shortest route program. Their shortest routes turn out to have total distances of 2193.698, 2145.632, 2131.345 and 2204.101.

The realtor calculates the CTSP multivariate correlation coefficient to be CTSP = A/B = 1606.458/((22193.698 + 2145.632)/2) = .7404, where B is the median of the four random shortest routes. Then taking the twelve A_i/B_i quotients using all combinations of the four random data shortest routes, we find the range of these quotients to be .9670 to 1.034. Now the majority of the sampling distribution of CTSP, if H_0 is true, should be in this region. However, the calculated CTSP = .7404 with the real data. Therefore, the null hypothesis (H_0) of no correlation between the variables can be confidently rejected.

Therefore, the variables are highly correlated because the shortest route connecting the real data is relatively short. This happens when the data is following a pattern driven by the correlation between the variables. It turns out that the Equation (5) fits the data quite well and can be used perhaps for forecasting sales volume in future months.

$$X_5 = .67\exp((X_1 + X_2 + X_3 + X_4)/80) \qquad (5)$$

THE CTSP STATISTIC

The CTSP statistic idea appears to be fairly general for finding a correlation between variables. A relatively shorter shortest route connecting the data points (when compared with the shortest routes of the similar amounts of random data in the same ranges) indicates that the data is more compact and following a pattern. This is fairly easy to picture in two and three dimensions (representing two and three variables). Think of a three dimensional arch. Points on the arch will have a much shorter shortest route than random three dimensional points in the same region that are not following a pattern.

A two dimensional example could be points that are on a circle. They will have a much shorter shortest route than the same number of points randomly scattered in the region near the circle and enclosed by it. Therefore, CTSP can pick up correlations that are not functions, as a circle is a relation and not a single valued function.

The past four years (2002-2006) have seen an intensive effort to see if this CTSP idea works in higher dimensions, four five, six, seven…variable problems when the geometry is difficult to picture. The success rate is very high. Almost every time CTSP finds correlation when it exists and does not find it when the variables are not correlated. Therefore, what is what is needed now is a software package to quickly do the CTSP analysis of the variables represented by the columns of the data and all appropriate subsets of these variables when the initial test results warrant further analysis.

The multi stage Monte Carlo optimization (MSMCO) TSP algorithm that is used on the CTSP analysis calculates the distance from each data point in *k* dimensions (for *k* variables represented by *k* columns of *n* rows of data) to each other data point. Then the distances are column ranked (one column for each of the *n* lines of data) from smallest to largest. Therefore, in this ranked array, the entry in the upper left hand corner would be the distance of the point closest to point one from it. The number below it would be the distance of the second closest point to point one and so on down the first column. The entry in the lower left hand corner would be the distance from point one to the point furthest away from it. Columns 2 through *n* would have the similar ranked distances for the rest of the points.

Then the MSMCO TSP simulation starts in the high end of this ranked array where a lot of the nearest points are. This makes the simulation much more efficient and able to close in on a good approximate shortest route quickly for all the A and B calculations needed for CTSP = A/B.

CONCLUSION

There is so much computing power available in our computer age of the twenty-first century that even extremely complex multivariate problems are within reach, especially if an approximate solution is acceptable. Multi stage Monte Carlo optimization is useful for approximating the solution to most any optimization problem. Examples of this are in (Conley 1993) and (Conley 1994). The CTSP correlation statistic is but one of several TSP statistics that are discussed in (Conley 2005). The CTSP correlation statistic can pick up any type of correlation (linear, nonlinear and even correlations that are not described by a function). Therefore, they can compliment the standard linear correlation analysis (Anderson 2003) that is commonly used today.

REFERENCES

Anderson, T.W. 2003. *An Introduction to Multivariate Statistical Analysis*, 3rd edition, Wiley and Sons, New York.

Conley, W.C. 1993. "Multi Stage Monte Carlo Optimization Applied to a Six Hundred Point Traveling Salesman Problem." *International Journal of Systems Science*, vol. 24, 309-626.

Conley, W.C. 1994. "Multi Stage Monte Carlo Optimization and Nonlinear Test Problems." *International Journal of Systems Science*, vol. 25, 155-171.

Conley, W.C. 2005. "Managing Large Data Sets with the TSP Statistics." In *Proceedings of the Modern Technology in the Innovation Processes of the Industrial Enterprises* (Genoa, Italy, Sept. 8-9). DIPTEM, Genoa, 191-196.

BIOGRAPHY

William C. Conley is a professor of business administration and statistics at the University of Wisconsin at Green Bay, U.S.A. He has a B.A., M.A., M.Sc. and Ph.D. in mathematics and computer statistics. The developer of multi stage Monte Carlo optimization and the TSP statistics, he has more than 185 publications worldwide. He is a senior member of the Society for Computer Simulation International and was named to *Who's Who in America* and the Albion College Athletic Hall of Fame in 1995 and 2005. He is a member of the American Chemical Society and a fellow in the Institute of Electronics and Telecommunication Engineers. He is a Michigan Scholar in College Teaching and was selected for the 2006 edition of Marquis *Who's Who in the World*. He teaches advanced business statistics to undergraduate students and does computer simulation research on multi stage Monte Carlo optimization (MSMCO) and the TSP class of statistics.

INVARIANT MANIFOLDS OF COMPLEX SYSTEMS

Jean-Marc Ginoux and Bruno Rossetto
P.R.O.T.E.E. Laboratory
I.U.T. de Toulon, Université du Sud
B.P. 20132, 83957, La Garde cedex,
France
E-mail: ginoux@univ-tln.fr, rossetto@univ-tln.fr

KEYWORDS
Invariant curves, invariant surfaces, multiple time scales dynamical systems, complex systems.

ABSTRACT

The aim of this work is to establish the existence of invariant manifolds in *complex systems*. Considering *trajectory curves* integral of multiple time scales dynamical systems of dimension two and three (predator-prey models, neuronal bursting models) it is shown that there exists in the phase space a *curve* (*resp.* a *surface*) which is invariant with respect to the flow of such systems. These invariant manifolds are playing a very important role in the stability of complex systems in the sense that they are "restoring" the determinism of *trajectory curves*.

DYNAMICAL SYSTEMS

In the following we consider a system of ordinary differential equations defined in a compact E included in ℝ :

$$\frac{d\vec{X}}{dt} = \vec{\Im}\left(\vec{X}\right) \qquad (1)$$

with $\vec{X} = \left[x_1, x_2, ..., x_n\right]^t \in E \subset ℝ^n$

and $\vec{\Im}\left(\vec{X}\right) = \left[f_1\left(\vec{X}\right), f_2\left(\vec{X}\right), ..., f_n\left(\vec{X}\right)\right]^t \in E \subset ℝ^n$

The vector $\vec{\Im}\left(\vec{X}\right)$ defines a velocity vector field in E whose components f_i which are supposed to be continuous and infinitely differentiable with respect to all x_i and *t*, i.e., are C^∞ functions in E and with values included in ℝ , satisfy the assumptions of the Cauchy-Lipschitz theorem. For more details, see for example (Coddington and Levinson 1955). A solution of this system is an *integral curve* $\vec{X}\left(t\right)$ tangent to $\vec{\Im}$ whose values define the states of the dynamical system described by the Equation (1). Since none of the components f_i of the velocity vector field depends here explicitly on time, the system is said to be autonomous.

TRAJECTORY CURVES

The integral of the system (1) can be associated with the co-ordinates, i.e., with the position, of a point M at the instant *t*. The total derivative of $\vec{V}\left(t\right)$ namely the instantaneous acceleration vector $\vec{\gamma}\left(t\right)$ may be written, while using the chain rule, as:

$$\vec{\gamma} = \frac{d\vec{V}}{dt} = \frac{d\vec{\Im}}{d\vec{X}}\frac{d\vec{X}}{dt} = J\vec{V} \qquad (2)$$

where $\dfrac{d\vec{\Im}}{d\vec{X}}$ is the functional jacobian matrix *J* of the system (1). Then, the *integral curve* defined by the vector function $\vec{X}\left(t\right)$ of the scalar variable t representing the trajectory of M can be considered as a *plane* or a *space curve* which has local metrics properties namely *curvature* and *torsion*.

Curvature

The *curvature*, which expresses the rate of changes of the tangent to the *trajectory curve* of system (1), is defined by:

$$\frac{1}{\Re} = \frac{\left\|\vec{\gamma} \wedge \vec{V}\right\|}{\left\|\vec{V}\right\|^3} \qquad (3)$$

where $\Re$ represents the *radius of curvature*.

Torsion

The *torsion*, which expresses the difference between the *trajectory curve* of system (1) and a *plane* curve, is defined by:

$$\frac{1}{\Im} = -\frac{\dot{\vec{\gamma}} \cdot \left(\vec{\gamma} \wedge \vec{V}\right)}{\left\|\vec{\gamma} \wedge \vec{V}\right\|^2} \qquad (4)$$

where $\Im$ represents the *radius of torsion*.

LIE DERIVATIVE – DARBOUX INVARIANT

Let φ a C^1 function defined in a compact E included in □ and $\vec{X}\left(t\right)$ the integral of the dynamic system defined by (1). The Lie's derivative is defined as follows:

$$L_{\vec{X}}\varphi = \vec{V}\cdot\vec{\nabla}\varphi = \sum_{i=1}^{n}\frac{\partial\varphi}{\partial x_i}\dot{x}_i = \frac{d\varphi}{dt} \qquad (5)$$

Theorem 1:

An *invariant curve* (*resp. surface*) is defined by $\varphi\left(\vec{X}\right)=0$ where φ is a C^1 in an open set U and such there exists a C^1 function denoted $k\left(\vec{X}\right)$ and called cofactor which satisfies

$$L_{\vec{X}}\phi\left(\vec{X}\right)=k\left(\vec{X}\right)\phi\left(\vec{X}\right) \qquad (6)$$

for all $\vec{X}\in U$

Proof of this theorem may be found in (Darboux 1878)

Theorem 2:

If $L_{\vec{X}}\varphi = 0$ then φ is first integral of the dynamical system defined by (1). So, φ is constant along each trajectory curve and the first integrals are drawn on the level set $\left\{\varphi=\alpha\right\}$ and where α is a constant.

Proof of this theorem may be found in (Demazure 1989)

INVARIANT MANIFOLDS

According to the previous theorems 1 and 2 the following proposition may be established.

Proposition 1: *The location of the points where the local curvature of the trajectory curves integral of a two dimensional dynamical system defined by (1) vanishes is first integral of this system. Moreover, the invariant curve thus defined is over flowing invariant with respect to the dynamical system (1).*

Proof of this theorem may be found in (Ginoux and Rossetto 2006)

Proposition 2: *The location of the points where the local torsion of the trajectory curves integral of a three dimensional dynamical system defined by (1) vanishes is first integral of this system. Moreover, the invariant surface thus defined is over flowing invariant with respect to the dynamical system (1).*

Proof of this theorem may be found in (Ginoux and Rossetto 2006)

APPLICATIONS TO COMPLEX SYSTEMS

According to this method it is possible to show that any dynamical system defined by (1) possess an *invariant manifold* which is endowing stability with the *trajectory curves*, restoring thus the loss determinism inherent to the non-integrability feature of these systems. So, this method may be also applied to any complex system such that predator-prey models, neuronal bursting models….
But, in order to give the most simple and consistent application, let's focus on two classical examples:

- the Balthazar Van der Pol model
- the Lorenz model.

Van der Pol model

The oscillator of B. Van der Pol, (1926) is a second-order system with non-linear frictions which can be written:

$$\ddot{x}+\alpha(x^2-1)\dot{x}+x=0$$

The particular form of the friction which can be carried out by an electric circuit causes a decrease of the amplitude of the great oscillations and an increase of the small. There are various manners of writing the previous equation like a first order system. One of them is:

$$\begin{cases}\dot{x}=\alpha\left(x+y-\dfrac{x^3}{3}\right)\\ \dot{y}=-\dfrac{x}{\alpha}\end{cases}$$

When α becomes very large, *x* becomes a "fast" variable and *y* a "slow" variable. In order to analyze the limit $\alpha\to\infty$, we introduce a small parameter $\varepsilon=1/\alpha^2$ and a "slow time" $t'=t/\alpha=\sqrt{\varepsilon}t$. Thus, the system can be written:

$$\vec{V}\begin{pmatrix}\dfrac{dx}{dt}\\ \dfrac{dy}{dt}\end{pmatrix}=\vec{\Im}\begin{pmatrix}f\left(x,y\right)\\ g\left(x,y\right)\end{pmatrix}=\begin{pmatrix}\dfrac{1}{\varepsilon}\left(x+y-\dfrac{x^3}{3}\right)\\ -x\end{pmatrix} \qquad (7)$$

with ε a positive real parameter: $\varepsilon=0.05$ and where the functions *f* and *g* are infinitely differentiable with respect to all x_i and *t*, i.e., are C^∞ functions in a compact E included in □ ^{2} and with values in □ .

According to Proposition 1, the location of the points where the local *curvature* vanishes leads to the following equation:

$$\phi\left(x,y\right)=9y^2+\left(9x+3x^3\right)y+6x^4-2x^6+9x^2\varepsilon \qquad (8)$$

According to Theorem 1 (Cf. Appendix for details), the Lie derivative of Equation (8) may be written:

$$L_{\vec{X}}\phi\left(\vec{X}\right)=Tr[J]\phi\left(\vec{X}\right)+\frac{2x^2}{\varepsilon}\left(-3x-3y+x^3\right)^2 \quad (9)$$

Let's plot the function $\phi(x,y)$ (in blue), its Lie derivative $L_{\vec{X}}\phi\left(\vec{X}\right)$ (in magenta), the *singular approximation* $x+y-\frac{x^3}{3}$ (in green) and the *limit cycle* corresponding to system (7) (in red):

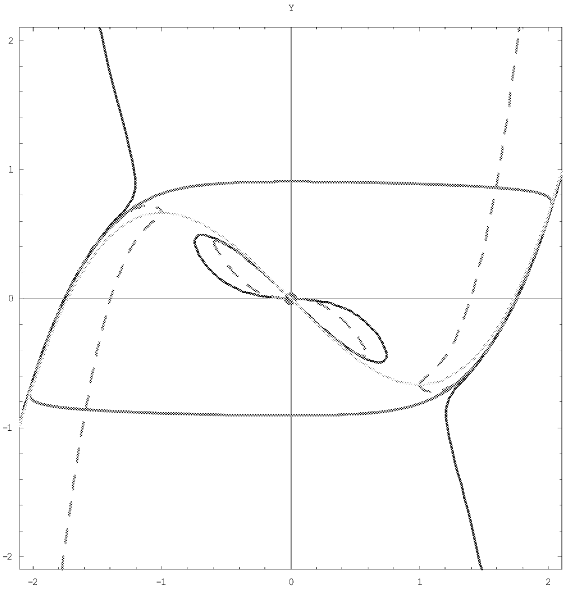


Figure 1: Van der Pol model

According to Fenichel's theory, there exists a function $\phi(x,y)$ defining a manifold (curve) which is overflowing invariant and which is $C^r O(\varepsilon)$ close to the *singular approximation*. It is easy to check that in the vicinity of the *singular approximation* which corresponds to the second term of the right-hand-side of Equation (9) we have:

$$L_{\vec{X}}\phi\left(\vec{X}\right)=Tr[J]\phi\left(\vec{X}\right)$$

Moreover, it can be shown that in the location of the points where the local *curvature* vanishes, i.e., where $\phi(x,y)=0$ Equation (9) can be written:

$$L_{\vec{X}}\phi\left(\vec{X}\right)=0$$

So, according to Theorem 1 and 2, we can claim that the manifold defined by $\phi(x,y)=0$ is an *invariant curve* with respect to the flow of system (7) and is a *local first integral* of this system.

Lorenz model

The purpose of the model established by Edward Lorenz (1963) was in the beginning to analyze the impredictible behaviour of weather. It most widespread form is as follows:

$$\vec{V}=\begin{pmatrix}\frac{dx}{dt}\\ \frac{dy}{dt}\\ \frac{dz}{dt}\end{pmatrix}=\vec{\Im}\begin{pmatrix}f(x,y,z)\\ g(x,y,z)\\ h(x,y,z)\end{pmatrix}=\begin{pmatrix}\sigma(y-x)\\ -xz+rx-y\\ xy-\beta z\end{pmatrix} \quad (10)$$

with σ, r, and β are real parameters: $\sigma=10$, $\beta=\frac{8}{3}$, r = 28 and where the functions *f*, *g* and *h* are infinitely differentiable with respect to all x_i, and *t*, i.e., are C^∞ functions in a compact E included in □ ^{3} and with values in □ . According to Proposition 1, the location of the points where the local torsion vanishes leads to an equation which for place reasons can not be expressed. Let's name it as previously:

$$\phi(x,y,z) \quad (11)$$

According to Theorem 1 (Cf. Appendix for details), the Lie derivative of Equation (11) may be written:

$$L_{\vec{X}}\phi\left(\vec{X}\right)=Tr[J]\phi\left(\vec{X}\right)+P\left(\vec{V},\vec{\gamma}\right) \quad (12)$$

where P is a polynomial function of both vectors $\vec{V}$ and $\vec{\gamma}$. Let's plot the function $\phi(x,y,z)$ and its Lie derivative $L_{\vec{X}}\phi\left(\vec{X}\right)$ and the *attractor* corresponding to system (10):

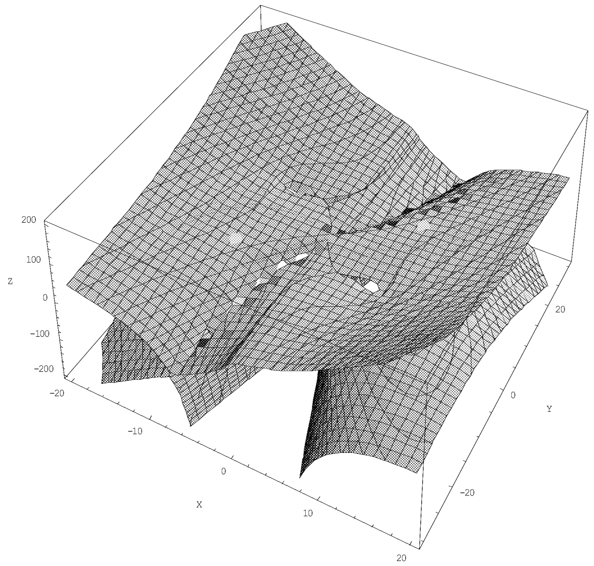


Figure 2: Lorenz model

It is obvious that the function $\phi(x,y,z)$ defining a manifold (surface) is merged into the corresponding to its Lie derivative. It is easy to check that in the vicinity of the manifold $\phi(x,y,z)$ Equation (12) reduces to:

$$L_{\vec{X}}\phi(\vec{X}) = Tr[J]\phi(\vec{X})$$

Moreover, it can be shown that in the location of the points where the local torsion vanishes, i.e., where $\phi(x,y,z)=0$ Equation (12) can be written:

$$L_{\vec{X}}\phi(\vec{X}) = 0$$

So, according to Theorem 1 and 2, we can claim that the manifold defined by $\phi(x,y,z)=0$ is an *invariant surface* with respect to the flow of system (10) and is a local first integral of this system.

DISCUSSION

In this work, existence of *invariant manifolds* which represent local first integrals of two (resp. three) dimensional dynamical systems defined by (1) has been established.
From these two characteristics it can be stated that the former implies that such manifolds are representing the stable part of the trajectory curves in the phase space and from the latter that they are restoring the loss determinism inherent to the non-integrability feature of such systems. Moreover, while considering that dynamical systems defined by (1) include complex systems, it is possible to apply this method to various models of ecology (predator-prey models), neuroscience (neuronal bursting models), molecular biology (enzyme kinetics models)… Research of such invariant manifolds in coupled systems or in systems of higher dimension (four and more) would be of great interest.

ACKNOWLEDGEMENTS

Authors would like to thank Professors M. Aziz-Alaoui and C. Bertelle for their useful collaboration.

REFERENCES

Coddington, E.A. & Levinson., N., 1955.
Theory of Ordinary Differential Equations,
Mac Graw Hill, New York.

Darboux, G. 1878. "Mémoire sur les équations différentielles algébriques du premier ordre et du premier degré".
Bull. Sci. Math. Sér. 2 (2), 60-96, 123-143, 151-200.

Demazure, M. 1989. *Catastrophes et Bifurcations*,
Ellipses, Paris.

Fenichel, N. 1979. "Geometric singular perturbation theory for ordinary differential equations". J. Diff. Eq. 31, 53-98

Ginoux, J.M. and Rossetto B. 2006. "Invariant manifolds of complex systems". *to appear.*

Lorenz, E. N. (1963) "Deterministic non-periodic flows,"
J. Atmos. Sc, 20, 130-141.

Van der Pol, B. 1926. "On 'Relaxation-Oscillations',"
Phil. Mag., 7, Vol. 2, 978-992.

APPENDIX

First of all, let's recall the following results:

$$L_{\vec{X}}\|\vec{u}\| = \frac{d\|\vec{u}\|}{dt} = \frac{\vec{u}\cdot\dot{\vec{u}}}{\|\vec{u}\|} \qquad (A-1)$$

Two-dimensional dynamical system

Let's pose: $\varphi(\vec{X}) = \|\vec{\gamma}\wedge\vec{V}\|$.

According to (A-1) the Lie derivative of this expression may be written:

$$L_{\vec{X}}\varphi(\vec{X}) = \frac{d\|\vec{\gamma}\wedge\vec{V}\|}{dt} = \frac{(\vec{\gamma}\wedge\vec{V})\cdot\frac{d}{dt}(\vec{\gamma}\wedge\vec{V})}{\|\vec{\gamma}\wedge\vec{V}\|} \qquad (A-2)$$

where $\frac{d}{dt}(\vec{\gamma}\wedge\vec{V}) = \dot{\vec{\gamma}}\wedge\vec{V}$

According to Equation (2) the Lie derivative of the acceleration vector may be written:

$$\dot{\vec{\gamma}} = J\vec{\gamma} + \frac{dJ}{dt}\vec{V} \qquad (A-3)$$

it leads to:

$$\begin{aligned}\frac{d}{dt}(\vec{\gamma}\wedge\vec{V}) = \dot{\vec{\gamma}}\wedge\vec{V} &= \left(J\vec{\gamma}+\frac{dJ}{dt}\vec{V}\right)\wedge\vec{V} \\ &= J\vec{\gamma}\wedge\vec{V}+\frac{dJ}{dt}\vec{V}\wedge\vec{V}\end{aligned} \qquad (A-4)$$

Using the following identity:

$$(J\vec{a})\wedge\vec{b} + \vec{a}\wedge(J\vec{b}) = Tr(J)(\vec{a}\wedge\vec{b})$$

it can be established that:

$$J\vec{\gamma}\wedge\vec{V} = Tr(J)(\vec{\gamma}\wedge\vec{V})$$

So, expression (A – 2) may be written:

$$\begin{aligned}L_{\vec{X}}\varphi(\vec{X}) = \frac{1}{\|\vec{\gamma}\wedge\vec{V}\|}(&Tr(J)(\vec{\gamma}\wedge\vec{V})\cdot(\vec{\gamma}\wedge\vec{V}) \\ &+\left(\frac{dJ}{dt}\vec{V}\wedge\vec{V}\right)\cdot(\vec{\gamma}\wedge\vec{V}))\end{aligned} \qquad (A-5)$$

Let’s notice that: $\left(\vec{\gamma} \wedge \vec{V}\right) \cdot \left(\vec{\gamma} \wedge \vec{V}\right) = \left\| \vec{\gamma} \wedge \vec{V} \right\|^2$ and that: $\vec{\beta} = \dfrac{\vec{\gamma} \wedge \vec{V}}{\left\| \vec{\gamma} \wedge \vec{V} \right\|}$

So, equation (A – 5) leads to:

$$L_{\vec{X}}\varphi\left(\vec{X}\right) = Tr\left(J\right)\left\| \vec{\gamma} \wedge \vec{V} \right\| + \left(\frac{dJ}{dt}\vec{V} \wedge \vec{V} \right) \cdot \vec{\beta} \qquad (A – 6)$$

Since vector $\dfrac{dJ}{dt}\vec{V} \wedge \vec{V}$ has a unique co-ordinate according to the $\vec{\beta}$-direction and since we have posed: $\varphi\left(\vec{X}\right) = \left\| \vec{\gamma} \wedge \vec{V} \right\|$, expression (A – 6) may finally be written:

$$L_{\vec{X}}\varphi\left(\vec{X}\right) = Tr\left(J\right)\varphi\left(\vec{X}\right) + \left\| \frac{dJ}{dt}\vec{V} \wedge \vec{V} \right\| \qquad (A – 7)$$

Three-dimensional dynamical system

Let’s pose: $\varphi\left(\vec{X}\right) = \dot{\vec{\gamma}} \cdot \left(\vec{\gamma} \wedge \vec{V}\right)$. The Lie derivative of this expression may be written:

$$L_{\vec{X}}\varphi\left(\vec{X}\right) = \frac{d\left[\dot{\vec{\gamma}} \cdot \left(\vec{\gamma} \wedge \vec{V}\right)\right]}{dt} \qquad (A – 8)$$

According to $\dfrac{d}{dt}\left[\dot{\vec{\gamma}} \cdot \left(\vec{\gamma} \wedge \vec{V}\right)\right] = \ddot{\vec{\gamma}} \cdot \left(\vec{\gamma} \wedge \vec{V}\right)$, it leads to:

$$L_{\vec{X}}\varphi\left(\vec{X}\right) = \frac{d\left[\dot{\vec{\gamma}} \cdot \left(\vec{\gamma} \wedge \vec{V}\right)\right]}{dt} = \ddot{\vec{\gamma}} \cdot \left(\vec{\gamma} \wedge \vec{V}\right) \qquad (A – 9)$$

The Lie derivative of expression (A – 3) leads to:

$$\ddot{\vec{\gamma}} = J\dot{\vec{\gamma}} + 2\frac{dJ}{dt}\vec{\gamma} + \frac{d^2J}{dt^2}\vec{V}$$

Thus, expression (A – 9) reads:

$$\begin{aligned} L_{\vec{X}}\varphi\left(\vec{X}\right) &= \left(J\dot{\vec{\gamma}}\right) \cdot \left(\vec{\gamma} \wedge \vec{V}\right) \\ &+ \left(2\frac{dJ}{dt}\vec{\gamma} + \frac{d^2J}{dt^2}\vec{V} \right) \cdot \left(\vec{\gamma} \wedge \vec{V}\right) \end{aligned} \qquad (A – 10)$$

It can also be established that:

$$\left(J^2\dot{\vec{\gamma}}\right) \cdot \left(\vec{\gamma} \wedge \vec{V}\right) = Tr\left(J\right)\left(J\dot{\vec{\gamma}}\right) \cdot \left(\vec{\gamma} \wedge \vec{V}\right)$$

So, since we have posed: $\varphi\left(\vec{X}\right) = \dot{\vec{\gamma}} \cdot \left(\vec{\gamma} \wedge \vec{V}\right)$, expression (A – 10) may finally be written:

$$\begin{aligned} L_{\vec{X}}\varphi\left(\vec{X}\right) &= Tr\left(J\right)\varphi\left(\vec{X}\right) + (-Tr\left(J\right)\frac{dJ}{dt}\vec{V} \\ &+ J\frac{dJ}{dt}\vec{V} + 2\frac{dJ}{dt}\vec{\gamma} + \frac{d^2J}{dt^2}\vec{V}) \cdot \left(\vec{\gamma} \wedge \vec{V}\right) \end{aligned} \qquad (A – 11)$$

JEAN MARC GINOUX was born in Toulon, France and went to the University of Nice Sophia-Antipolis, at I.N.L.N. of Nice and then at the University of South, where he studied nonlinear and chaotic dynamical systems with Professor Bruno Rossetto. He obtained his Ph-D in Applied Mathematics in 2005. He worked for a couple of years for the Department of Education before moving in 2004 to I.U.T. of Toulon where he has been working in Dynamical Systems ever since.

GIS
AND
COMPLEXITY

THE EVOLUTION PROCESS OF GEOGRAPHICAL DATABASE WITHIN SELF-ORGANIZED TOPOLOGICAL PROPAGATION AREA

Hakima Kadri-Dahmani[1] Cyrille Bertelle[2] Gérard H.E Duchamp[1] Aomar Osmani[1]

[1] Lipn-University Paris 13
99 avenue Jean-Baptiste Clément, 93430 Villataneuse, France
E-mail: {hkd, ao,Gerard.Duchamps @lipn.univ-paris13.fr

[2] LITIS - University of Le Havre,
25 rue Philippe Lebon, BP 540, 76058 Le Havre cedex, France
E-mail: cyrille.bertelle@univ-lehavre.fr

KEYWORDS
geographical database; complex systems; evolution; emergence.

ABSTRACT

The paper deals with Geographical Data Base evolution which is a major aspect of the actual development in Geographical Information System (GIS). In a more practical aspect, GIS has now to evolve to manage updating. We will explain how the updating processes can be described as an evolution processus for GIS and make them transform from complicated systems to complex systems.

INTRODUCTION: GIS and their evolution process

A Geographic Information System (GIS) is a computer-based tool using geographical objects. A GIS is composed of a Geographical Data Base (GDB) with applicative operators which allow it to get, to stock, to verify, to manipulate, to analyze and to represent the spatial data of the GDB.

The originality of Geographical Data Base from ordinary Data Base is the use of spacial data (Rigaux, 2002). The latter may be represented in a Geographical Data Base with two aspects: in raster mode or in vector mode. The raster mode is based on a pixels grid representation. The vector mode manipulates geographical features. In the following, we will focus our attention on the vector mode representation where each feature is represented by an object. Each object has a semantic part describing the nature of a feature which it represents and a geometric part describing its shape and its localisation. The position of the objects the ones compared with the others is an important information which is usually represented in the Geographical Data Base.

So, in Geographical Data Bases, geographical information is often represented with three levels: geometric, semantic and topological. From each level, we can define relations between objects that have to be linked corresponding to the specific level.

At the semantic level, a Geographical Data Base is often structured with layers. Layers are generally defined concerning a specific thematic like road traffic, fluvial tracing, building or vegetation. Generally, objects of a same layer have the same geometric representation and share the same topological properties inside networks.

The constant evolution of the real world which must be represented in the geographical data base induces the need of regularly update of the GDB and so make it evolve (Kadri-Dahmani, 2002). To develop automatic evolution processes of a GDB, we must introduce inside the GDB itself, an adapted data representation containing relation between the objects.

In this paper, we present how GIS under evolution process can be shown as a complex system. In section (3) we explain how the objects of the geographical data base interact at the time of an update. A crucial problem emerges then: which are the objects of the base concerned?

We propose in section (4) a propagation algorithm which allow tthe propagation of interactions. From this algorithm emerges a property given in the section (5). Section (6) presents implementations and experiments and we conclude in section (7).

FROM GIS TO COMPLEX GIS

All these structured informations which defines a GIS introduce a great number of static dependences but each layer can be generally understood alone or some parts of each layer can be isolated to better understand the dependence between involved objects. Generally the applicative operators can be computed on each of these parts. In that way, we can consider classical GIS as complicated systems in the terminology proposed by Le Moigne (Lemoigne, 1999). We can consider that the Geographical Data Base in association with the previous applicative operators which constitute the GIS is a closed system.

Today, the complexity of the world needs to use or to add additional functionalities on GIS. Geographical informations deal also with human-landscape interactions. The simulation of social aspects and of ecological processes seems to be more and more linked to the better understanding of the geographical data and its evolution inside its all social, geopolitic and ecological environment. To integrate these new aspects, we have to manage some complex processes like some energetic fluxes that crosses the standard GIS (see the figure 1). This complex fluxes transform the standard GIS in open systems which confer to them some properties linked to complexity. Self-organization and multi-scale organizations can emerge from these complex processes. The expected evolutions of GIS can be considered as the transition which will transform the standard complicated GIS into complex GIS. We say that GIS is under an updating flow.

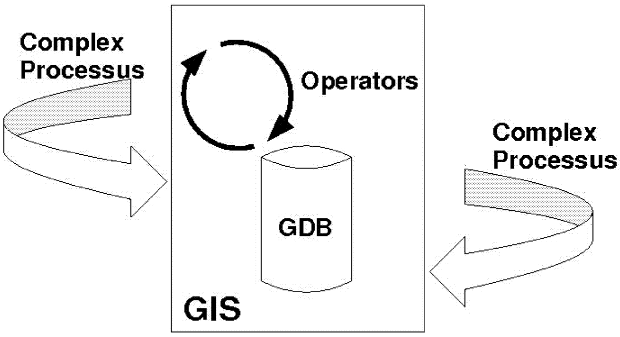


Figure 1: GIS under Complex Processus (updating flow).

In the following, we deal with a specific improvement on GIS which concerns its own evolution. As described in the previous section, the constant evolution of the real world induces the need to regularly update the geographical data of GIS. This evolution processus is typically a complex processus that generate some dynamical organizational processes inside GIS. The data themselves retro-act on the processus during the propagation method that we will present in the next sections (Kadri-Dahmani, 2005).

EVOLUTIVE GIS FORMALISM

We adopt the feature-based approach, where features are the fundamental concept for the representation of geographical phenomena as described in (Kadri-Dahmani, 2005).
Basically, a GDB is represented in a minimal formalism, by the pair (V, D) where:

1. V is the set of the classes used in the GDB. Each class gathers features which have common characteristics. The set V gathers GDB scheme elements.

2. D is the definition domain of the variables of V. It is the set of the objects of one GDB instance. The heterogeneity of objects which belong to a GDB needs their classification for a better use. We consider 4 classes of objects which are spread over two information levels: the geometric level which gathers geometric primitives PG and the geometric objects OG, and and the semantic level which gathers simple semantic objects OS and complex semantic objects OC.

$$D = PG \cup OG \cup OS \cup OC$$

The proposed model for a GDB which allows to evolve through updating operations must add some complementary sets which will manage some dependences between the geographical elements. So the model will be composed of a quadruplet (V, D, R, C) where, in addition:

3. The connection graph over the GDB elements is based on relations between these elements. R is the set of these relations. The different kinds of relations that we consider are: composition relations RC, dependence relations RD and topologic relations RT

$$R = RC \cup RD \cup RT$$

In this paper, we focus our study on topological relationships. Topological relations are fundamental and allow to describe the relative position of the objects with each others. There is many models which propose a representation of this kind of relation, we propose to use the 9-intersection model from Egenhofer and Herring (Egenhofer, 1991). In this model, each object p_i is defined by the inset, noted p°i , the outline set, noted δpi, and the exterior set, noted p^{-}_{i} . This model can be represented by a matrix formulation. A topological relation between two objects X and Y is represented by a matrix. Each element of this matrix denotes the intersection between different components of X and Y (Egenhofer, 1991).

4. C is a set of constraints defined between the variables of V and/or between variables value of V. In our object modelization, this corresponds to constraints defined between the classes (constraints between variable) and/or between objects (constraints between values). These constraints manage the GDB evolution on many levels.
This quadruplet corresponds to the GDB modelisation to prepare it to evolution.

5. Finally, to effectively manage evolution processes, we have to modelize the updating informations in accordance with the GDB conceptual model. We note M the updating set where basis action is the transaction and the full model for the GDB is the 5-uplet (V, D, R, C, M)

Example 1: Figure 2 shows a geometric representation of a Round-about object O_i. O_i is a complex object composed of 5 simple objects. Each two simple objects are connected by a topological relation "touch".

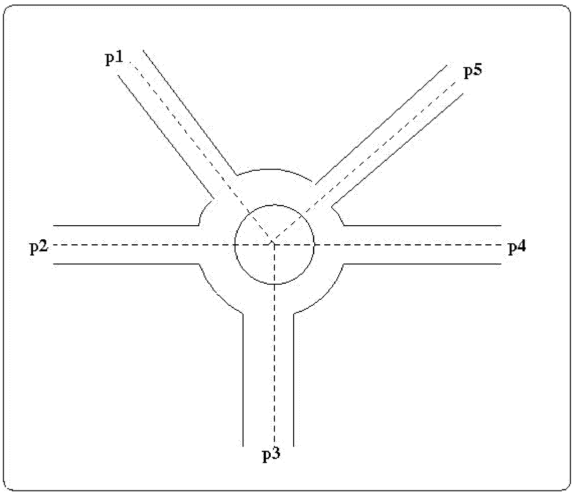


Figure 3: Round-about object geometric representation.

UPDATING PROPAGATION : Dynamic Interaction Network

The integration of the updating flow in the geographical data base starts the dynamics of the system evolution. Let's recall that updating flow is structured as a set of transactions, each one is a set of canonical operations sequences. So, each canonical operation, applied on an object, activates some falls of updating operations to apply on other objects. At the topological level, geographical objects are linked with topological relations. The evolution of an object X in topological relation with the object Y, needs also the evolution of this last one. The evolution of Y will need also to make evolve the other objects in relation with it. Usually the evolution of X, called starting operation, is known since it is a membership of the input updating flow, but its influence on the objects linked to X is not known. So, consequent operations, called influence operations, are not known. Objects to which these operations are applied, called influenced objects, are not either known.

So, propagate the effect of an updating operation in the geographical data base means to execute the starting operation and all the influence operations which result from it without altering its consistency. This is translated, in our system, by the installation of an interaction network built from the connection graph and the table of influences in (Kadri-Dahmani, 2005).

If we consider an object Oc and an associated updating operation, op, the mecanism of propagation is applied in a local zone centered on the object Oc, called working zone from where we extract the set of other objects which may be under the influence of the first one. The propagating mechanism is recursif but we limit the exploration inside the working zone. The step is presented in the following algorithm where InfluenceAreaT() alow to delimit the Topological Influence Area defined as follow:

Definition 1. We consider Oc a target object to be updated from a GDB. Its Topological Influence area that we note as Zit(Os), collects all the GDB objects which are inside a working zone of Oc and which are linked to 0c.

Example 2 Figure 5 shows working zone of the target object ID 9814 which is a camping. The table1 contains a part of the set of the ID objects which belong to his influence area. For each ID object, its corresponding class and the topological relation with the target object are given.

We represent the topological influence area by a connection graph which connects the objects linked by a topological relation.

Example 3. Figure 6 shows some elements of the connection graph corresponding to some relation from the target object ID 9814, based on the table 1.

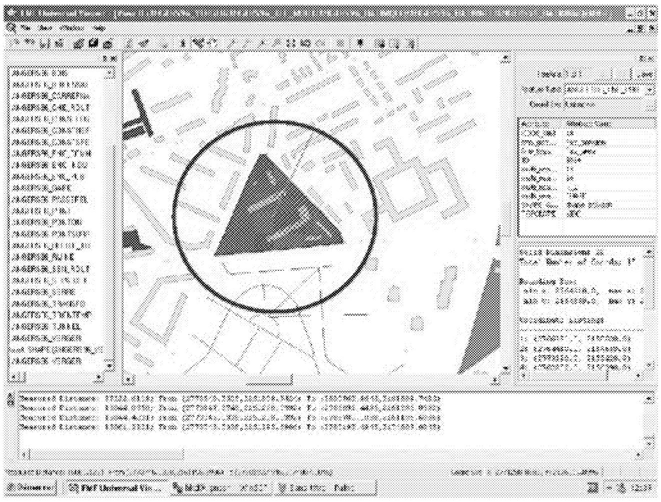


Figure 5: Topological influence area from object ID 9814

ID	CLASS	RELATION
7130	BATIMQCQ	contains
7135	BATIMQCQ	contains
7134	BATIMQCQ	contains
3640	ROUTE TR	borders
2264	ROUTE TR	borders
4520	CARREFOURNA	touchs
4530	CARREFOURNA	touchs
...	....	...

Table 1: Topological influence area from object ID 9814

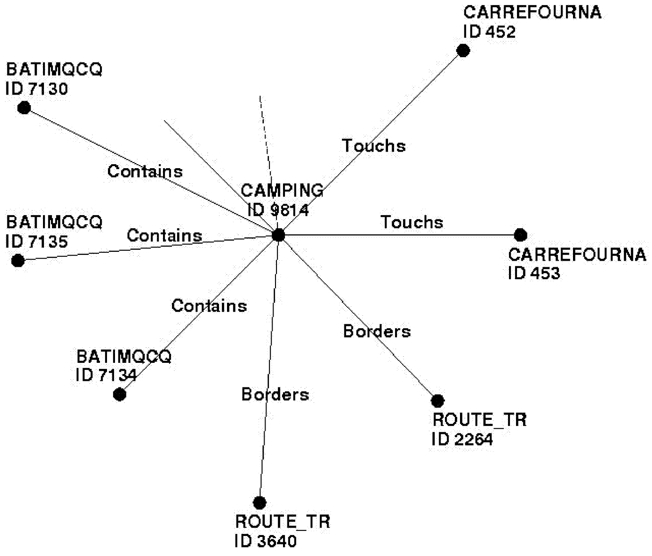


Figure 6: Connection graph (extract from object ID 9814)

```
Propagate(Op1, O1, O2, Rel, tabu, Result, Zte,BC):Result
If Op1=Identity or O2=Null or Rel=Null
then
    if $O2 \neq Null$
    then tabu ← tabu – {O2}
    else tabu ← tabu – {O1}
    endif
    Result ← true
else
    Op2 ←InfluenceTableVisit (Op1, O1, O2, Rel)
    Zitc(O2) ← InfluenceAreaT(O2, Zte).Zitc(O2)
    if DirectUpdate (Op2, O2, Zitc(O2),BC).Success
    then Result ← Result $\cup$ {O2}
        Success ← true
        For all object $O3 \in$ Zitc(O2) then
            if ($O3 \in\notin$ tabu)
            then
                Rel2 ← Relation(O2,O3)
                Success ← Propagate (Op2, O2, O3, Rel2, tabu $\cup$ {O3}, Result, Zte, BC}
            endif
        endfor
    else Success ← False
    endif
endif
return success
```

EMERGENT PROPERTY

The local propagation allows to avoid to explore the whole geographic data base. Now, we need to build a processus that will compute the adapted working zone which permits to tell whether the local consistency maintenance is enough to insure the global consistency maintenance. For that purpose we propose an algorithm that we called dilatation method and that consist to progressively increase the working area like a new disk centered on the initial object Oc and which radius is augmented step by step with the value p until that a further increase will not compute new objects involved in the propagation processus. That leads us to define this computed area as the stability area associated to the object Oc. To prove that the local consistency maintenance can be sufficient for the global consistency maintenance, we had to define some hypothesis about the regularity of the objects repartition. The properties given in the following are prove in (Kadri-Dahmani, 2005).

Definition 2. A finite set of planar points is called p-dense if the Delaunay triangulation over all the set of points has no edges longer than p.

Property 1. If the influence area of the point Oc is p-dense then the dilatation method with a step equal to p computed from Oc give the stability area of this point.

Property 2. The local consistency over the stability area for an object Oc will insure the consistency of the whole Data Base.

This last emergent property allows us to define a subset of objects from the GDB and be able to predict that the behavior of these objects is himself the behavior of all objects of the GDB vis-a-vis to a flow of update.

This first theoretical result allows us to implement in an efficient way, the whole updating system, with the postulate that the natural geographical data follows this hypothesis of regular distribution, using an adapted step of resolution for the dilatation method.

IMPLEMENTATION AND EXPERIMENTS

The whole system has been originally developed in the COGIT laboratory where it has been implemented. This system is in operational practice and has been connected to the framework OXYGENE (Badard, 2003) of this laboratory. OXYGENE, mainly developed in Java, allows to model and to use geographical information within oriented object scheme. Data Bases are managed with Oracle which has been completed with a spacial data extension. The mapping between Oracle and Object representation is made by JDO (Java Data Object) and OJB.

A methodology for its validation has been developed and has proved that the mechanisms are efficient, even if some rejection can be avoid with a better scheduling. An experiment has been developed on the IGN GDB concerning the Angers French town zone (see figure 7).

Using some matching technique between two of these GDB from 1994 and 1996 (respectively called BDTopo94 and BDTopo96), we have built a set of updating informations. We define 2 indicators to validate the automatic updating process respecting the consistency maintenance:

- The precision indicator defined as the rate of correct updating decision number over the total realized decision number;
- The similarity indicator which compares the updating results from BDTopo94 with the true situation in BDTopo96. This indicator is the rate of the number of matched objets in this comparison over the total number of matched objects and unmatched objects.

From a specific experiment based on 30 canonical operations, we have obtained a precision indicator equal to 0.948 and a similarity indicator equal to 0.962 (Kadri-Dahmani, 2005) which is sufficient to validate the whole processus.

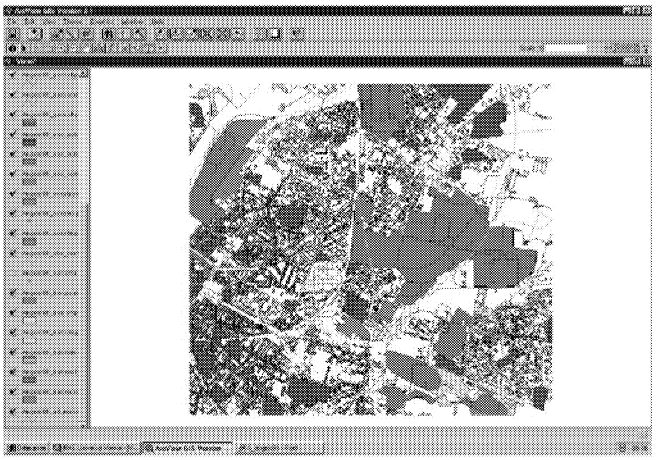


Figure 7: Extract of BDTopo96 concerning Angers town

CONCLUSION

This paper describes a consistent updating processus over a Geographical Data Base (GDB) as a complex operation concerning Geographic Information Systems (GIS). Our purpose is to explain where the complexity occurs during the processus, The formalism proposed for the geographical information description is based on objects. To manage the updating, we have to define semantic and topological relations which allow to define the influence area associated with each object. These relations between objects are represented with a connection graph.

Moreover, we have to manage some constraints which deal with consistency maintenance of the whole data base ; a specific language is proposed for that purpose. Even if the connection graph is important on a whole geographic map, the previous system to describe GIS is complicated in the sense that we can manage it correctly by successive splitting and application of basic operators.

The updating processus is then defined over the GIS as a complex flux that make involved the GIS. This processus implements a propagation method which consists to act by updating on GDB objects and these objects propagate the updating operators using an influence table and so retro-acts on the whole system and processus. In that sense, the updating processus crosses the GIS like an evolutive organizational flux which transform the GIS from a complicated system to a complex system. The basis of the updating is the use of influences tables which summarize all the canonical operators needed. The application of this influence table can be compared with the rule based processus which make involved a cellular automaton.

Finally we show how the updating processus, as a complex flux over GIS, can lead to obtain an emergent property. This property allows to obtain the global consistency maintenance of the whole GDB from only local consistency maintenance. We implement a dilatation method that can be considered as a way to obtain a self-organization concerning the updating problem.

The complex decomposition and description of the work presented in this paper allows us to build conceptual models over GIS which can be used to manage some others kinds of complex processes. We can adapt this proposed method for updating flux to other kinds of complex processes flux. In these complex processes, we can consider the human aspects of geography which deal with social, geopolitic and ecological purposes (6). The proposedmethods used here can give conceptual approaches to manage such major developments which give all the power in the use of GIS in our present complex world, to better understand and analyze it.

REFERENCES

Badard, T. and Braun, A. (2003) “OXYGENE: an open framework for the deployment of geographic web services” in Proceedings 21th International Cartographic Conference, Durban, South Africa.

Egenhofer, M. and Herring, J. R. (1991) “Categorizing Binary Topological Relations Between Regions, Lines and Points in Geographic Databases”, Technical Report, Departement of Surveying, University of Maine.

Kadri-Dahmani, H. (2001) “Updating in GIS: Towards a more generic approach” in Proceedings 20th International Cartographic Conference, Beijing, China.

Kadri-Dahmani, H. and Osmani, A. (2002), “Updating Data in GIS: How to maintain Database Consistency?” in Porceeding 4th International Conference of Entreprise Information system”, Ciudad Real.

Kadri-Dahmani, H. (2005) “Mise à jour des Bases de données géographiques et maintien de leur cohérence”, Ph. D. Thesis, University of Paris 13.

Le Moigne, J.-L. (1999) “La modélisation des systèmes complexes”, Dunod.

Rigaux, P., Scholl, M. and Voisard, A. (2002) “Spatial Databases – with applications to GIS”, Morgan Kauffmann.

SELF-ORGANIZATION SIMULATION OVER GEOGRAPHICAL INFORMATION SYSTEM BASED ON MULTI-AGENT PLATFORM

Rawan GHNEMAT, Cyrille BERTELLE
LITIS - EA 4051
Le Havre University
25 rue Philippe Lebon, BP540
76 058 Le Havre cedex, France
rawan.ghnemat@gmail.com
cyrille.bertelle@univ-lehavre.fr

Gérard H.E. DUCHAMP
LIPN - UMR CNRS 7030
Paris XIII University
99 avenue Jean-Baptiste Clément
93430 Villetaneuse, France
gheduchamp@gmail.com

KEYWORDS
GIS, Self-organization, Individual-based model (IBM), Agent-based modeling and simulation (ABMS)

ABSTRACT

In this paper, we present a review concerning the coupling of Geographical Information System with agent-based simulation. With the development of new technologies and huge geographical databases, the geographers now deal with complex interactive networks which describe the new Geopolitics and world-wide Economy. The aim is to implement some self-organization processes that can emerge from these complex systems. We explain how we can today model such phenomena and how we can implement them in a practical way.

1. INTRODUCTION

The challenge of the simulation of complex systems modelling based on Geographical Information System (GIS) is to propose the future decision making supports for urban plannings or environmental, social-politic development. Geographers manage today a great amount of geographical data and need innovative methodologies to analyse them in a pertinent way. A complex vision of the current world is strongly needed in order to face nowadays challenges in understanding, modelling and simulating (Aziz-Alaoui and Bertelle, 2006). The increasing of intensive communications, allowed by high technologies, leads to develop in efficient way, the information access and sharing. This revolution deeply transforms Geopolitics and world-wide Economy into geographical complex systems within huge interactive networks. Mixing GIS with complexity modelling is a new challenge to understand, analyse and build decision support systems for Economy and Geopolitics.

The actual GIS conceptual model is based on a layered structure (Goodchild et al., 1991). A layer allows gathering, in a same set, some objects corresponding to a specific thematic, hydrological layer, building layer, road layer, ... Some new challenges for GIS future development, is to use the new conceptual models given by complexity theories to automatically identify some dynamical organizations and so to manage scenario of developments which can be used to determine automatically some social or urban re-organizations, for example.

The use of Individual-based Model (IBM) offers potential for studying complex system behaviors and human/landscape interactions within a spatial framework. Artificial intelligent agents introduce behavior conditions and set communications or interactions between them and their environment as the major rule of the simulation evolution. Agents have goals who lead their interactions or actions over the world. Few researchers have mixed spatially explicit agents and GIS. After a non exhaustive review on that subject, we will focus our attention on Agent Analyst, based on Repast Multiagent system. Agent Analyst can be presented as a free extension to ESRI's products like ArcGIS.

2. MIXING INDIVIDUAL-BASED MODELS AND GIS

The use of Individual-Based Model (IBM) is a promising approach to model spatially explicit ecological phenomena. Interest has increased in using GIS for simulation to spatial dynamic processes. A great part of the challenge of modeling interactions between natural and social processes has to do with the fact that progress in these systems results in complex temporal-spatial behavior (Gimblett, 2002).

The successive efficient computer science concepts used for that purpose:

- Cellular automata theory which has demonstrated efficiency about modeling landscape dynamics by regular rules-based processes;
- Object-oriented modeling (OOM) which has proved its power by representing the domain with concrete objects that have as much similarity with their real world counterparts as possible;
- The spatially explicit model mixed with object-based modeling lead to define the individual-based modeling (IBM). The characteristics and advantages of IBM are (Gimblett, 2002):
  - A variety of types of differences among individuals in the population can be accommodated;
  - Complex system decision making by an individual can be simulated;
  - Local interactions in space and the effects of stochastic temporal and spatial variability are easily handled.

- Artificial intelligent agents (Wooldridge, 2002) introduce behavioral conditions and set communications or interactions between them and their environment as the major rule of the simulation evolution. Agents have goals who lead their interactions or actions over their world.

Few researchers have mixed spatially explicit agents and GIS. The first goal is to have a virtual laboratory to study the outcomes of various behavior on realistic landscapes.

Jiang and Gimblett provide an example of modeling pedestrian movements using virtual agents in urban spaces. Corresponding to cognitive processes for their behavior, agents act relatively to their own environment perception. Decision makers, such resources managers or designer, faced with realistic environment problems, would substantially benefit from these simulation techniques.

Itami and Gimblett have developed the Recreation Behavior Simulator (RBSim) to simulate the behavior of human individuals in natural environments. RBSim joins two computer technologies: GIS to represent environment and Autonomous Human Agents to simulate human behavior within geographic space. This tool can allow us to investigate tourism management options. It tries to give answers to some questions like: how different management options might affect the overall experience of tourists? How schedule some visits and know the impact on the number and frequency of users?

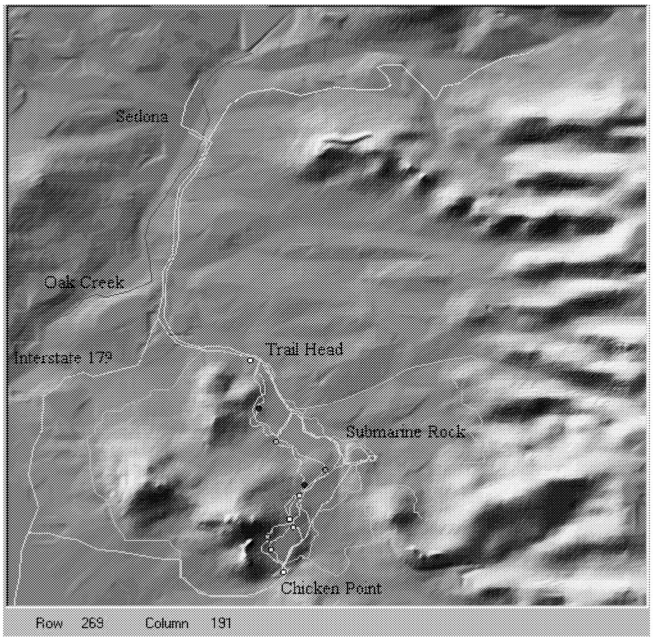


Figure 1: Dynamical agents move across the landscape in RBSim (after Gimblett)

Randy Gimblett develops studies about Colorado River through the Grand Canyon national park. This natural site suffers from impacts of repeated recreational uses like campsites and the destruction of sensitive vegetation due to many visitors. A management plan has been established to protect the national resource. RBSim is currently used to simulate the current use pattern of commercial tour operators in the Grand Canyon. In addition, managers wish to test alternative management strategies including alternative timetables for starting trips, booking specific riverside campsites for different sized boating parties. These strategies aimed of reducing conflicts between different boating parties, reducing environmental impact on river beaches, and increasing the number of river users.

After this brief review, we will present in the next section, a self-organized spatial model which will be implemented in a mixing GIS (Geographical Information System) - ABMS (Agent Based Modeling and Simulation) model for urban development.

3. SCHELLING'S SELF-ORGANIZED SEGREGATION MODEL

Thomas Schelling received in October 2005, the Nobel Price in Economic Sciences. He contributed to enhance the understanding of conflict and cooperation about social institutions. His major work deals with game theory, he proposes a simple model of spatial segregation which can lead to self-organized phenomena. Schelling's city segregation model illustrates how spatial organizations can emerge from local rules, concerning the spatial distribution of people which belong to different classes. In this model, people can move, depending on their own satisfaction to have neighbours of their own class. Based on this model, a city can be highly segregated even if people have only a mild preference for living among people similar to them.

In this model, each person is an agent placed on a 2D grid (in his original presentation, a chessboard was used by Thomas Schelling). Each case can be considered like a house where the agent lives. Each agent cares about the class of his immediate neighbours who are the occupants of the abutting squares of the chessboard. Each agent has a maximum of eight possible neighbour. Each agent has a "happiness rule" determining whether he is happy or not at his current house location. If unhappy, he either seeks an open square where his happiness rule can be satisfied or he exits the city. The rule-based system is described as following:

- An agent with only one neighbour will try to move if the neighbour is of a different class than his own;
- An agent with two neighbours will try to move unless at least one neighbour is of the same class as his own;
- An agent with from three to five neighbours will try to move unless two neighbours are of the same class as his own;
- An agent with from six to eight neighbours will try to move unless at least three neighbours are of the same class as his own.

The exact degree of segregation that emerges in the city depends strongly on the specification of the agents happiness rules. It is noticeable that, under some rule specifications, Schelling's city can transit from a highly integrated state to a highly segregated state in response to a small local disturbance. We can observe some bifurcation phenomena which lead to chain reaction of displacements.

In the figure 2, we present some results of Schelling's model computed on a cellular automaton based on the applications proposed by N. Gessler (Gessler, 2006).

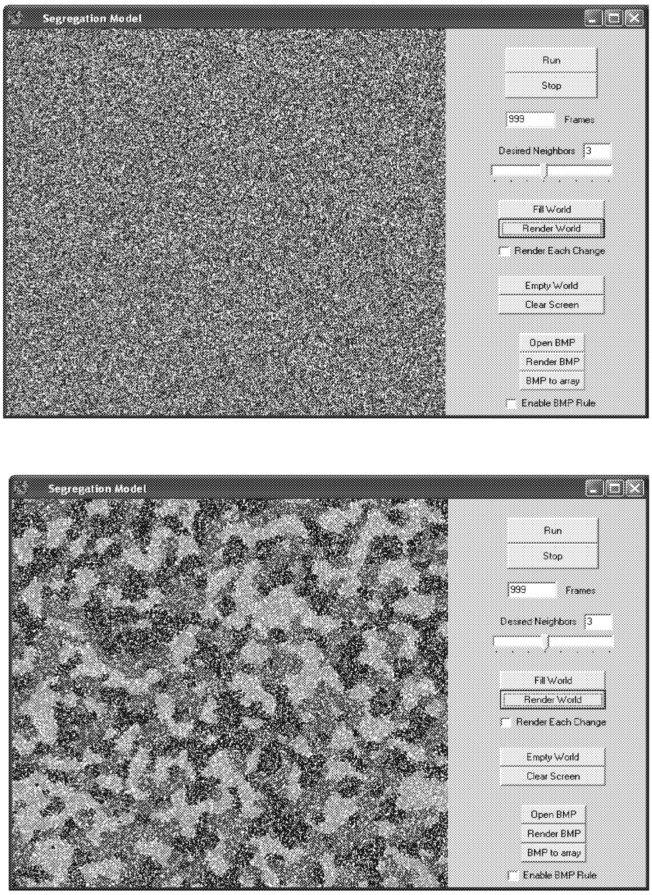


Figure 2: Schelling's segregation model implemented by N. Gessler (initial random ditribution and aftr 999 steps)

4. MIXING AGENT-BASED SIMULATION AND GIS MODEL SUPPORT TO IMPLEMENT SELF-ORGANIZED SEGREGATION MODEL.

Agent Analyst is an Agent-Based Modeling and Simulation (ABMS) extension for the ESRI's ArcGIS suite of products (Agent Analyst, 2006). Agent Analyst integrates and extends the functionalities of the open-source Repast modeling and simulation. The Recursive Porous Agent Simulation Toolkit (Repast) is an agent modeling toolkit (Repast, 2006). It borrows many concepts from the Swarm agent-based modeling toolkit, but it multiple pure implementations in several languages and built-in adaptive features such as genetic algorithms and regression. Repast seeks to support the development of extremely flexible models of living social agents, but is not limited to modeling such entities.

Agent Analyst fully integrates ABMS with GIS. Through this integration, GIS experts gain the ability to model behaviours and processes as change and movement over time (e.g., simulate land use and land cover changes, predator-prey interactions, or network flows and congestion) while ABMS modellers are able to incorporate detailed real-world environmental data, perform complex spatial processes, and study how behaviour is constrained by space and geography. Furthermore, ABMS models can include real-time GIS data for situations such as disaster management, fire fighting, or resource management.

The graphical Agent Analyst tools allow the user to create agents, schedule simulations, establish mappings to ArcGIS layers, and specify the behaviour and interactions of the agents. In the following, we will show how using Agent Analyst for Schelling's problem. A sample will be given using GIS and agent model and develop the use to implement it.

In the following, the Schelling segregation model is applied on a GIS vector layer rather than a 2D grid (Lingman-Zielinska, 2005). The implementation is made using Agent Analyst over ArcGIS. The model is composed of two main parts. The first one is an environment layer of zip code regions from a GIS and represented in Agent Analyst by a non movable vector GIS agent. The other part is a set of movable generic agents, which represent the city residents who make decisions of moving to a new location (zip region). This general architecture is represented by the figure 3 .

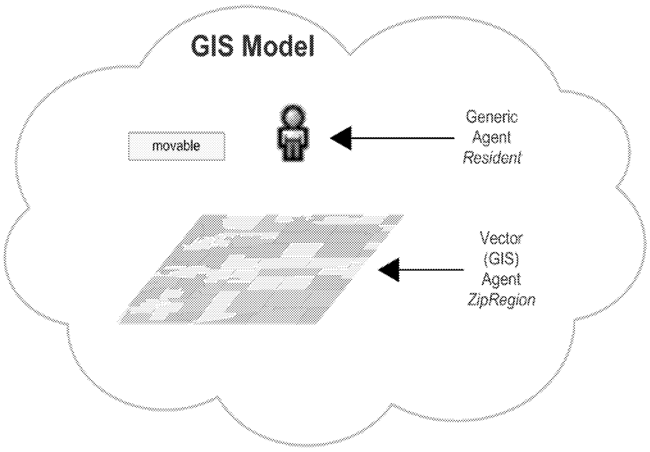


Figure 3: Agent-GIS mixing after (Agent Analyst, 2006)

The basis of the resident agent displacements come from the local rules of preference described below and a segregation phenomenon emerges from the interplay of these individual choices.

The model assumptions are based on a twofold distinction, red and green resident identity. The model is equipped with a basic parameter threshold called *tolerance*, which reflects agent preference level of neighboring agents that are alike (i.e. the same "color"). For example, if the tolerance parameter is set to 30%, then it is assumed that the agent will be satisfied if the ratio of identity-different neighboring agents to total number of neighbors does not exceed 30%. If it does, then the agent is dissatisfied and moves out to another region that is currently unoccupied.

Using Agent Analyst, we load the city layer shape in ArcMap as represent in figure 4.

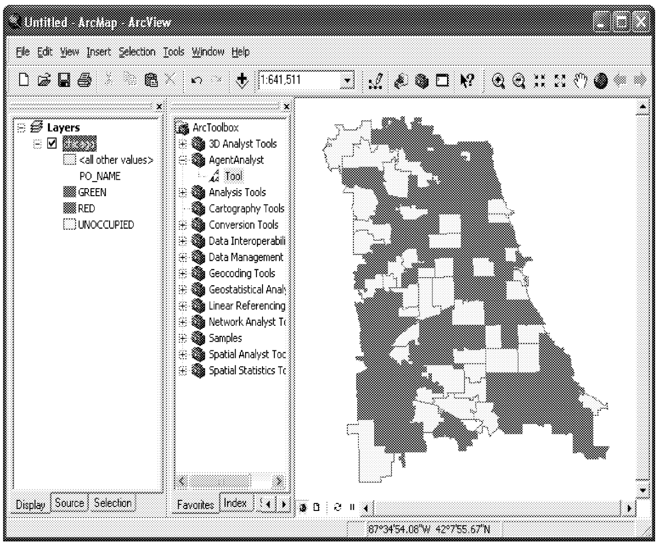
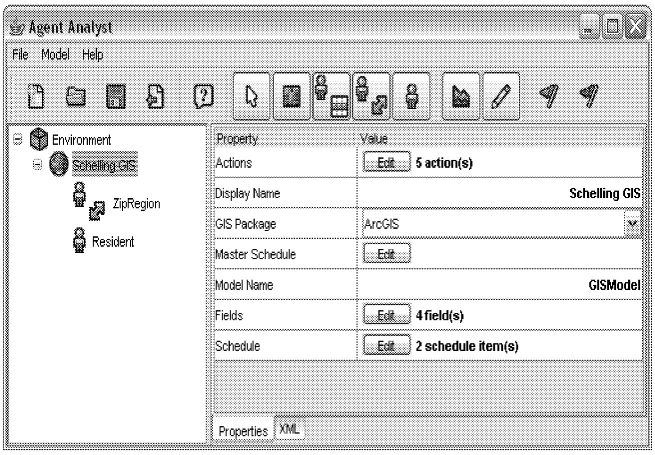


Figure 4: City layer Shapefile

Then the Schelling model is loaded in the ArcToolbox defined for AgentAnalyst as process. The figure 5 is the interface obtained. We can find on this interface a lot of elements like display name or GIS package used (ArcGis or OpenMap) but also Actions editor.



Property	Value
Actions	Edit 5 action(s)
Display Name	Schelling GIS
GIS Package	ArcGIS
Master Schedule	Edit
Model Name	GISModel
Fields	Edit 4 field(s)
Schedule	Edit 2 schedule item(s)

Figure 5: Agent Analyst model component.

Actions editor was designed to automate the process of programming model actions. It composes all the necessary elements that constitute *an action*. Since *an action* is a method in object-oriented programming language, which is here a variant of Python, need to define its name, import the necessary classes (modules) from outside the model (if we need any), use pre-existing variables, and finally write the code for action.

Once all the elements are defined, we can compile and run the model. One possible issue of the schelling model in the initial map defined previously can be represented by figure 6. In this figure we have the Repast toolbar which allows managing the simulation execution.

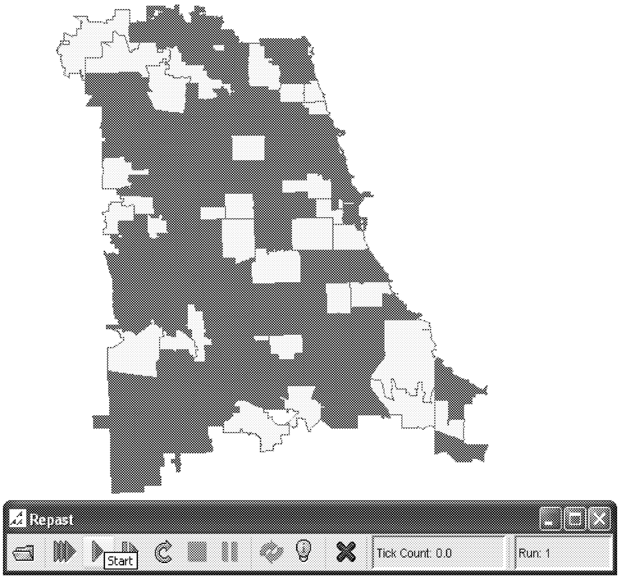


Figure 6: Agent Analyst simulation

Even if segregation is an important problem to manage or to control the urban systems development, the schelling model can be used and generalized to many others problems, especially for non structured networks. A. Singh and M. Haahr [48] have proposed an extension of Schelling model for managing peer to peer (P2P) networks. P2P is a new way of decentralized organization for communication over networks. Because of the increasing possibilities to share and to distribute efficient computers at low cost on many position on a network, the P2P way of communication could be one of the major in the future. The robustness of such networks in any kind of critical situations can make them powerful and efficient.

In an unstructured decentralized network, the location of the peers is decided randomly and therefore peers with high bandwidth may be adjacent to peers with low bandwidth introducing undesired low bottleneck bandwidth in the network. Schelling method is based on local displacement, which change the peers connections of the considered computer over the whole connection graph. Schelling method is also based on satisfaction criteria which is the fact that the neighbors have similar bandwidth than the considered computer. This demonstrate of Schelling's model can be used effectively for adapting P2P network topology and so to improve bandwidth usage. We can generalize this study to any kind of unstructured networks such as mobile telecom infrastructures distribution over an urban system area.

5. CONCLUSION AND PERSPECTIVES

We have presented in this paper, an implementation of the Schelling's segregation model over a Geographical Information System, using mixing with Agent Based Modeling and Simulation. Extension can be find to P2P network management but also more generally, to interaction network management. The mixed GIS-ABMS platform is intended now to be connected to ant systems (Bertelle et al., 2006) and automata models to allow to represent evolutive negociation processes (Ghnemat et al., 2006). Applications to economy and urban development are well-suited to this fore coming studies.

REFERENCES

Agent Analyst web site (2006) http://www.institute.redlands.edu/agentanalyst/

Aziz-Alaoui M.A. and Bertelle C. (2006) “Emergent Properties in Natural and Artificial Dynamical Systems”, Understanding Complex Systems, Springer.

Bertelle, C., Dutot, A., Guinand, F. and Olivier, D. (2006) "Organization Detection using Emergent Computing". *Int. Transactions on Systems Science and Applications*, Special Issue *"Self-Organizing, Self-Managing Computing and Communications"*.

Gessler, N. (2006) “”Building Complex Artificial Worlds” web site.

Ghnemat, R., Oqeili, S., Bertelle, C. and Duchamp, G.H.E. (2006) “Automata-Based Adaptive Behavior for Economic Modeling Using Game Theory” in M.A. Aziz-Alaoui and C. Bertelle (eds) “Emergent Properties in Natural and Artificial Dynamic Systems”, Understanding Complex Systems, Springer.

Gimblett, H.R ed. (2002) “”Integrating Geographic Information Systems and Agent-based Modeling Techniques”, Santa Fe Institute studies in the sciences of complexity, Oxford University Press.

Goodchild, M.F., Rhind, D. and Maguire, D.J. (1991) “Geographical Information Systems: Principles and Applications”, Longman, New York

Ligman-Zielinska, A. (2005) “Agent Analyst tutorial – Schelling GIS”, San Diego University.

Repast web site (2006) http://repast.sourceforge.net/

Singh, A. and Haar, M. (2004) “Topology adaptation in P2P Networks using Schelling's model”, Workshop on Emergent Behavior and Distributed Computing, PPSN-VIII

Wooldridge, M. (2002) “An introduction to MultiAgent Systems”, John Wiley & Sons.

Rawan GHNEMAT is a PhD student in LITIS laboratory of Le Havre University. She works in complex systems modelling for Geographical Information Systems. After a bachelor of science in geomatics and a master of science in computer science at Al-Balqa'Applied University in Jordan, she obtains a scholarship from French government to make a PhD in LITIS laboratory, Le Havre University, France.

Cyrille BERTELLE is professor in computer science in Le Havre University and develops research activities in complex systems modeling. His current activities concerns self-organization processes formalization within various applications: ecosystems and natural systems, decision making modeling based on GIS models, emotional modeling for decision making. He is one of the co-directors of a research laboratories amalgamation which promotes the Sciences and Technologies in Information and Communication over the Haute-Normandie in France.

Gérard H.E. DUCHAMP is one of the founders of the series of congresses FPSAC. Born in 1951 (Paris, France), he took his studies and degrees in the region of Ile de France and began trainer for the competitive examinations of "Grandes écoles". He received his Ph. D. and Habilitation in Paris VII under the direction of Dominique Perrin and Marcel-Paul Schützenberger (a member of french Academy of Sciences), both founders of the french school of Theoretical Computer Science. Pr.G.H.E. Duchamp's interests cover essentially the interplay between computation and the other areas of knowledge. His publications cover many domains where computation is involved such as: Automata Theory, Lie algebras, Quantum groups, Combinatorics (he made a video on the subject with Xavier Viennot), Computer algebra, Representation Theory and Quantum Physics.

CLIFF COLLAPSE HAZARDS SPATIO-TEMPORAL MODELLING THROUGH GIS : FROM PARAMETERS DETERMINATION TO MULTI-SCALE APPROACH

Anne DUPERRET
EA 2255 LMPG
Université du Havre,
25 rue Philippe Lebon, BP540
76 058 Le Havre cedex, France
anne.duperret@univ-lehavre.fr

Cyrille BERTELLE
EA 4051 LITIS
Université du Havre
25 rue Philippe Lebon, BP540
76 058 Le Havre cedex, France
cyrille.bertelle@univ-lehavre.fr

Pierre LAVILLE
BRGM
Maison de la Géologie
77 rue Claude Bernard
75 005 Paris, France
p.laville@brgm.fr

KEYWORDS

GIS, rule-based engine, multiscale modelling, geology, risk analysis, general systems theory, feedback

ABSTRACT

In this paper, we study the cliff collapses, using observations and in situ measures, along 120 km of the french chalk coastline in Upper-Normandy and Picardy. Cliff collapses occur inconsistently in time and space, in unpredictable way. A european scientific project ROCC (Risk Of Cliff Collapse) has been launched (1999-2002) in order to identify the critical parameters involved, to evaluate their impact and their interaction in mass movements. Cliff collapse process appears as a complex natural system, due to the large amount of parameters able to lead to a collapse. GIS approach has been used to allow an homogeneous cartography of each parameter reported on one layer each one, along a large surface of 120 km long coastline. The computation is decomposed in different steps which consist from the qualitative factors, to quantify them and to normalize them in space. On the basis of field measurements and data analysis, four types of geological information have been added to the GIS model and a first computation of hazard modelling has been proposed to design a collapse hazard sensitivity map, based on a elementary summation of the parameters. We now prospect to introduce a ruled-based systems, dealing with the complexity of the interaction of the local parameters. An interaction network must be defined to represent the spatial and semantic links between local parameters.

1. GEOLOGICAL CONTEXT

Coastal chalk cliffs exposures along each part of the English Channel are composed of nearly vertical cliffs ranging from 20 to 100 m high, with a less or more thin cover of clays-with-flints and a chalk shore platform with a low angle of slope, often covered with sand and/or shingles. The shore platform is made of eroded chalk and is subjected to a semi-diurnal cycle of macrotides, whereas the cliff rocks are submitted to fresh groundwater that infiltrate within the chalk through rainfall (Figure 1).

Coastal chalk cliffs exposed on each part of the English Channel suffer numerous collapses, with mean volumes varying between 10 000 and 100 000 cubic meters per event. Between October 1998 and October 2001, a minimum of 52 collapses have been observed along 120 km of the French chalk coastline located in Upper-Normandy and Picardy, with 28 collapses with volumes greater than 1000 cubic meters. Such collapses occurs inconsistently in time and space and appears to be relatively unpredictable. Little work has been devoted to the analysis of processes responsible for the collapses of the chalk seacliffs, and this led to the European scientific project, ROCC (Risk Of Cliff Collapse) because of the growing hazard to local communities from chalk cliff retreat. The main goal of the ROCC project was to identify the critical parameters leading to chalk coastal cliff collapses, and to evaluate the impact of those parameters and their interaction in such rock mass movements. The main objective was to create maps showing the sensibility of cliffs to erosion (cliff collapse hazard) along the 120 km coastline.

2. CLIFF COLLAPSE PROCESS

The evolution of a cliff from stability toward failure, depends on changes present in the rock mass (lithology, fracture pattern), and processes acting within the rock mass (degree of water saturation, water movement) caused by external agencies of subaerial and marine origin. External agencies lead to :

- the development and opening of fractures (resulting from stress relief, fatigue, wetting and drying, freeze-thaw action),
- the deterioration of the rock material as a result of infiltration of water (resulting in solution, chemical alteration, physical breakdown through freeze-thaw or salt crystallisation),
- substantial geometric changes at the cliff foot (height of shingles and debris accumulation).

The rock mass characteristics such as chalk type, fracture pattern and karstic development may be consider as fixed parameters, only varying in space. Variable parameters such as water saturation of the chalk and water movement in the chalk through fractures and karstic system are closely linked to external agencies, with various delays. External agencies are varying in space and time, with various fitting scales. It is the case of climatological parameters, such as rainfall and temperature (with temporal variations from hours, days, seasons, years and decade) and oceanographic parameters with temporal variations from day to season (for tides and wave action), and variations from year to decade (for sea level variation due to global change).

3. GIS APPROACH

Cliff collapse process appears as a complex natural system, due to the large amount of parameters able to lead to a collapse. GIS approach has been used to allow an homogeneous cartography of each parameter reported on one layer each one, along a large surface of 120 km long. GIS has been also used to perform various combinations between each parameters to obtain various degrees of hazard.

The coastline location has been identified on the IGN topographic cartography basis at 1/25 000 (MNT IGN©1992, as a raster). Each GIS layer is dedicated to one parameter. The most simplified method to attribute a level to each parameter is to evaluate the minimal and maximal value of the parameter within a fixed geographic framework. The operator attribute the level zero for the minimal value and the value 100 for the maximal value. Then, each parameter (with various original units) may be reported on a layer with percents as a value of the parameter intensity ; moreover parameters may be combined easily together on the same geographic framework. The report of each parameter has been realised on a coastal strip located at the top of the cliff (drawn from a 50 m wide dilatation each part from the coastline). For each parameter, various strip sections (polygons) are defined as a function of the parameter value.

The GIS framework is usefull for the spatial correlations of various parameters and to combine several models to represent an hazard level.

4. PARAMETERS DEFINITIONS

Georeferenced data produced by IGN have been used to build the GIS basis through Mapinfo® software. The various layers of the GIS are composed of parameters all varying in space, but non-variable or variable in time. Non-variable parameters in time are the geographic information (coastline location, cliff height) and the geologic information (chalk lithostratigraphic succession, fracturation). Variable parameters in time are the hydrogeologic and the oceanographic informations. Unfortunatly, oceanographic parameters have not been considered in this study. The cliff collapses occurrence results from the intercation of all these parameters, but we may consider that the observed location of past cliff collapses is a non variable parameter in time.

4.1 Coastline location and cliff height

One layer is dedicated to the spatial location of the coastline (precisely defined at the top of the cliff) to build a 100 m wide strip.

A second layer is dedicated to the cliff altitude and has been used to select various height sections (as polygons) within the strip, with 10 m intervals, varying from 5 to 100m height. These layers have been deduced from the 1/25 000ème IGN topographic map (not shown in this study).

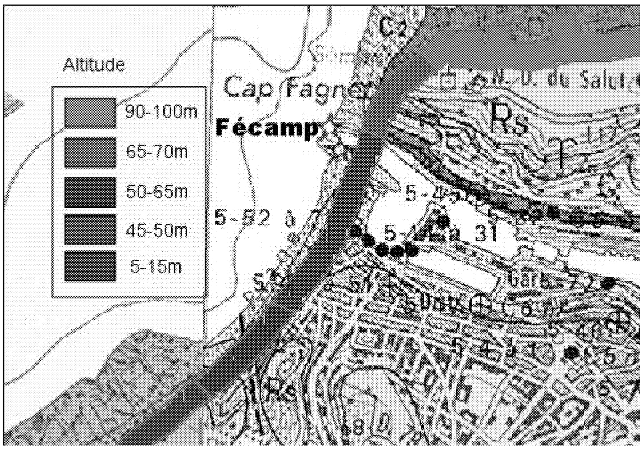


Figure 1 : Geological map (© BRGM) as raster around Fécamp and the strip indicative of the cliff altitude

4.2 Past cliff collapses occurrence

During the ROCC project, regular field surveys performed during october 1998 and december 2001 allowed to report a minimal value of cliff collapses. For each reported collapse, the location and date of occurrence were reported and the volume of the deposit were measured (Duperret et al., 2004). Such data have been reported on a layer in the GIS, as recent collapses parameter.

4.3 Chalk lithostratigraphy

As defined in UK (Mortimore, 1983), chalk type units are defined on the basis of a lithostratigraphic concept and are more representative of the geotechnical properties of the chalk than the stratigraphic scale traditionaly used in France (Mortimore, 2001 ; Duperret et al., 2004).

The Chalk lithostratigraphy layer is composed of six chalk units of various characteristics detailed below :

(1) Cenomanian craie de rouen is a nodular chalk with numerous flint bands in Upper-Normandy. Unit (1a) is defined between Antifer and Fécamp headlands, (1b) is defined north of Fécamp and (1c) south of Antifer.

(2) the Holywell nodular chalk is a nodular and massive chalk, with few flint bands, which contains many flaser marls and abundant Mytiloides shell debris layers, with open crossed fractures north of Fécamp (2a) and closed crossed fractures southward (2b)

(3) the New Pit chalk formation contains numerous flint bands in cliffs south of Fécamp (3a) but is flintless northwards at St Martin-plage, north of Fécamp (3b),

(4) The Lewes Nodular Chalk is a yellowish coarse chalk, including soft, marly bands and nodular hardgrounds, with regular flint layers. The Lewes Nodular Chalk formation contains dolomitic layers to the south of Fécamp (4a) which are absent northward (4b),

(5) The Seaford Chalk Formation is a homogeneous white chalk with conspicuous bands of large flints, with large collapses, north of Fécamp (5a) and small collapses southward (5b)

(6) The Newhaven Chalk Formation is a marly chalk with numerous marl seams and regular but few flint bands.

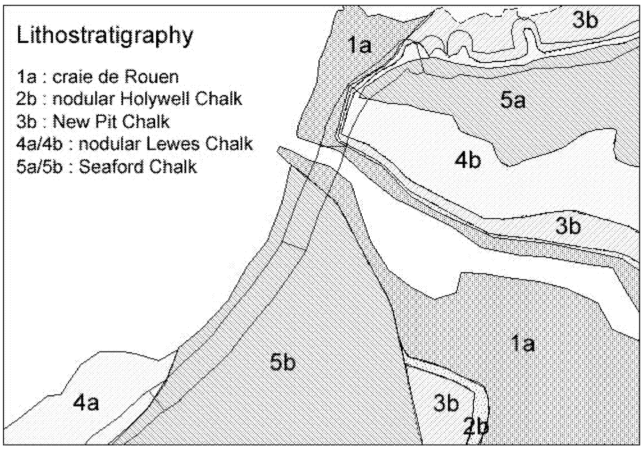


Figure 2 : chalk lithostratigraphy map, around Fécamp.

Chalk rocks present various degrees of physical properties, particularly density and porosity. The standart strength categories used to describe rocks for engineering purposes could not be applied readily to chalk, due to the variations in physical properties even within a single block of intact chalk.

Even if each chalk unit is well defined, a direct comparison of each chalk units is not realistic. Moreover, some types of chalk retain water at saturation level while others gain and lose water more readily, changing drastically their physical properties.

The lithostratigraphic indice has thus been determined by a direct correlation between each vertical lithological succession on the cliff face and the estimated volumetric mass able to collapse. Each chalk unit formation overlay an older one. On the cliff face, various chalk successions may appears, depending on the height of the cliff and the thickness of each chalk units (Fig. 3). From field works, chalk units succession have been recognized on the cliff face outcrops, conducting to the definition of twenty four vertical successions of chalk units (Mortimore, 2001, Duperret et al., 2004).

As examples : five chalk units outcrop on a vertical section at Fécamp (1a/2b/3b/5b/5a) whereas one chalk unit outcrops at Dieppe (6). Cliff collapses volumes have been estimated for each event observed in the field in France and UK. The involved volumes are varying from 1 to 100 000 m^3 (Mortimore et al., 2004). Seven volumic classes have been defined and each volumic class has been attributed to the corresponding lithostratigraphic sequence. On the basis of involved volumes during a collapse, a percent scale has been established for each lithostratigraphic sequence. The maximum influence has been established for the sequence 1a/2b/3b/4b/5a (at Fécamp) and sequence 4b/5a (100%), with collapses of mean volumes reaching 55 000m^3 and the minimum influence is established for the sequence 1b/2b (0%), with no reported collapse. The six other intermediates classes are based on a logarithmic scale and defined at 99%, 84%, 78%, 58%, 42% and 35%, with involved collapsed mean volumes of 50 000 m^3, 10 000 m^3, 5000 m^3, 500 m^3 and 100 m^3, 50 m^3 respectively.

4.4 Fracturation

On the basis of the observations in the field, a preliminary hypothesis was suggested that fractures embedded within the chalk cliffs could influence cliff collapse. About 2000 fracture orientation measurements were collected on 34 investigations sites regularly spaced along the 120 km long coastline. Fracture analysis were completed and homogenised on a systematic analysis of the cliff face from a continuous set of oblic aerial photography of the cliff face, realised in 1986. The correlation between field data and continuous aerial photography acquisition has been used to define two major layers dedicated to fracture occurrence (Genter et al., 2004).

i) Transverse fracturation indice

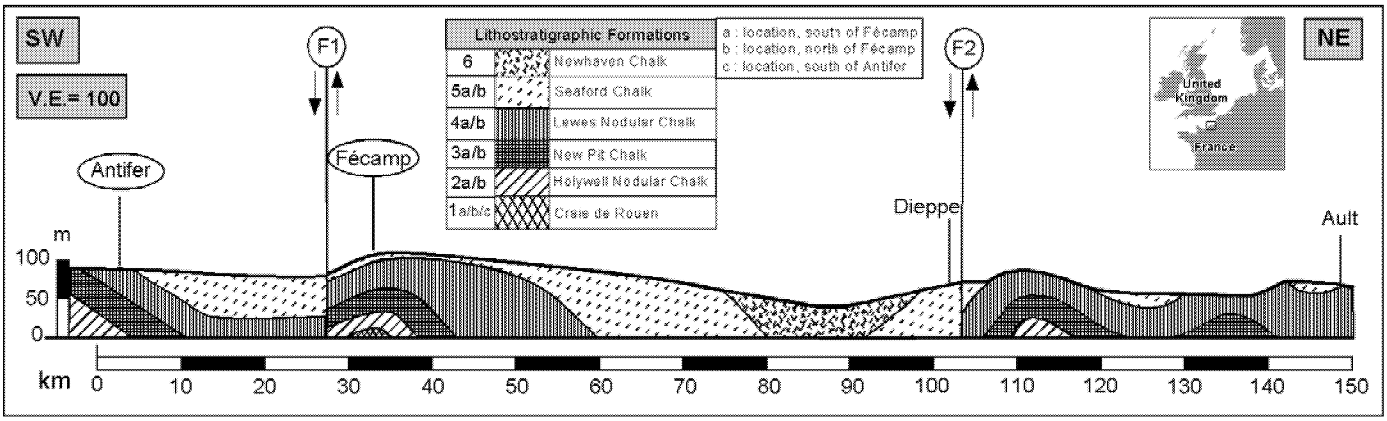
The total number of fractures that appear on the cliff face represents in fact the number of transverse fractures to the cliff face. From this numbering, various sections of fracture density have been defined. The total number of fractures reported on a complete area of the cliff face vary from zero to 396 per square meter, which is equivalent to a scale of fracture density varying from zero to 0.17 and a mean space between fractures varying from zero to 95.5. An indice of transverse fractures has been calculated from these data, varying from zero to 100% and has been used to define 63 sections of various length on the strip, with various degrees of transverse fracturation.

ii) Parallel fracturation indice

A second layer has been dedicated to fracture data parallel to the cliff face. Such data have been detected on aerial photographs on the naked beach platform, where fractures set

appears easily. Calculations have been realised on a density fracturation scale with four levelsreported in percent in the GIS.

Figure 3 : vertical cross section of the various chalk lithostratigraphic sequence on the cliff face



4.5 Hydrogeology

Experiments on chalk rocks show that hydration produces a marked decrease of the chalk strength, which varies depending on the chalk type. When chalk samples are submitted to a progressive water wetting, a fall of strength occurs. The decrease of the UCS strength is between 20 and 50% of the dry strength of chalk and this reduction begins with very low values of water content within the chalk (Duperret et al., 2005). Chalk rocks formations are said to exhibit a dual porosity/permeability. In a classic dual-porosity aquifer the matrix pores provide storage and fissures provide the permeable pathways for flow. At large scale, the chalk aquifer presents a behaviour of a porous system, with low flow and at small scale, the chalk aquifer presents a behaviour of fissural system with high flows.

At large scale, the water content of the chalk varies with fluctuations of groundwater level, submitted to rainfall inputs. The magnitude of the water table fluctuations in Upper Normandy is generally inversely proportional to the degree of fissuring (i.e. low permeability areas with less fractures have high water table fluctuations) (Crampon et al., 1993). Hydrogeology data have been summarized on three layers in the GIS (Caudron et al., 2001).

i) Water table level

Data have been deduced from the hydrogeological map at 1/100 000 edited by BRGM. Original data was compiled from various available piezometric data in upper eptembe, that were acquired at various step of space and time. Even if this information is unprecise, it is able to give some interesting trends concerning the water table location in depth. Four classes of water table have been defined as a function of chalk imbibition thickness near the coastline : (0) zero for suspended valleys, (1) low imbibition thickness (lower than 5 m), (2) moderate imbibition thickness (around 10 m), (3) high imbibition thickness (higher than 10 m), (4) very high imbibition thickness (coastal area covered by impervious tertiary cover). The water table indice is ranging from 2.5 to 100 %, with the lowest water table effect for the class (1) and the highest water table effect for the class (4).

ii) Coastal piezometric slope

Like water table layer, the coastal piezometric slope has been directly deduced from the hydrogeological map at 1/100 000. The degree of coastal piezometric slope gives an indication of the hydric flow, from the aquifer to the coastline. Five classes have been defined, with a piezometric slope ranging from 0, lower than 5 ‰, between 5 and 10 ‰, between 10 and 20 ‰, between 20 and 40 ‰, and higher than 40 ‰. The higher the slope is, the higher hydric flow is, the higher the influence to collapse is.

iii) Coastal springs occurrence

Coastal springs locations reveal mainly the occurrence of fissural and/or karstic system in the chalk. These data have been collected in the field during eptember-october 1999 (fall 1999). Three classes have been defined depending of the springs flow and the spring density on a coastal section : (1) no coastal springs or coastal springs with low flow (lower than 10 l/s) and low linear density, (2) coastal springs with low flow (lower than 10 l/s) and high linear density or coastal springs with mean flow (between 10 and 100 l/s) and low linear density, (3) coastal springs with mean flow (between 10 and 100 l/s) or coastal springs with high flow (higher than 100 l/s). As a first approximation, the higher the flow and density are, the higher the karstic system is developed, and the higher the fissural flow transit is.

5. HAZARD MODELLING

5.1 Arithmetic combination

Based on the GIS model, a first computation of hazard modelling is proposed. The goal is to design a collapse hazard sensitivity map. The computation is decomposed in different steps which consist from the qualitative factors, to quantify them and to normalize them in space. The computation is based on a elementary summation of the parameters. The resulting sum in each coastal strip must be considered as a potential level of low to high degree of collapse hazard, based on a percent deduced from all parameters. A part of the resulting GIS is presented in the figure 4. Nevertheless, a confrontation with models and observations needs to be performed to introduce weighting in association with the pertinent parameters.

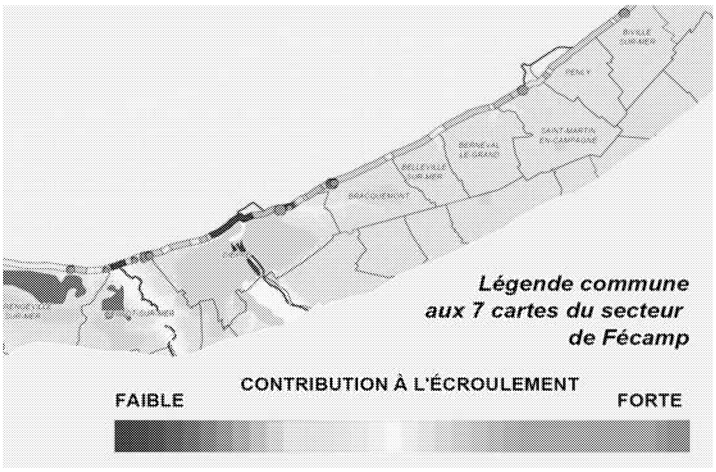


Figure 4 : Detail of the hazard sensitivity map deduced from arithmetic combination between each parameters

5.2 Multi-scale rule-based qualitative system

We now prospect to introduce a ruled-based systems, dealing with the complexity of the interaction of the local parameters. A process modeling allows to describe the alea estimation from complex interaction between geological structure, external agencies and hydrodynamic phenomena. The process modeling is described on Figure 5 where transitions are activate by physical qualitative modeling rules (Kuipers, 1986). During some specific external agencies (rainfall, temperature, ...), some links will be activated and dynamically propagate the phenomenon through the rules-based process which finally give in/output of the cliff collapse hazard. The resulting collapse hazard can be considered as a kind of emergent computation. This process concerns a local description and the multi-scale approach consists to change the level of description and to represent each local process as a compartment. The compartments are linked by spatial transfers that correspond to pressure variation and water displacement along the fracture network as represented by the figure 5. The implementation of the whole method is in progress.

REFERENCES

Duperret A., Genter A., Martinez A., Mortimore R.N., 2004, Coastal chalk cliff instability in NW France : role of lithology, fracture pattern and rainfall, *in* : Mortimore R.N. and Duperret A. (eds) *Coastal chalk cliff instability*. Geological society, London, Enginneering Geology Special Publications, 20, 33-55.

Duperret A., Taibi S., Mortimore R.N., Daigneault M., 2005, Effect of groundwater and sea weathering cycles on the strength of chalk rock from unstable coastal cliffs of NW France, *Engineering Geology*, 78, 321-343.

Caudron M., Equilbey E., Mortimore R.N., 2001, Projet ROCC : Hydrogéologie. Etat critique des connaissances et impact de l'eau sur la stabilité des falaises.

Crampon N, Roux J.C., Bracq P., Delay P., Lepiller F., Mary M., Rasplus L, Alcayadé G., 1993. France. *in* : Downing R.A., Price M and Jones G.P. (eds*). The Hydrogeology of the Chalk of North-West Europe*. Oxford Science Publications, 113-152.

Genter A., Duperret A., Martinez A., Mortimore R.N., Vila J.-L., 2004, Multiscale fracture analysis along the French chalk coastline for investigating erosion by cliff collapse, *in* : Mortimore R.N. and Duperret A. (eds) *Coastal chalk cliff instability*. Geological society, London, Engineering Geology Special Publications, 20, 57-74.

Kuipers B., 1986, Qualitative simulation, *Artificial Intelligence*, 29, 289-338.

Mortimore R.N., 1983. The stratigraphy and sedimentation of the turonian-Campanian in the southern Province of England. *Zitteliana*, 10, 27-41.

Mortimore R.N., 2001, Report on mapping of the chalk channel coast of France from Port du Havre-Antifer to Ault, June-September 2001, Bureau de Recherches Géologiques et Minières (BRGM).

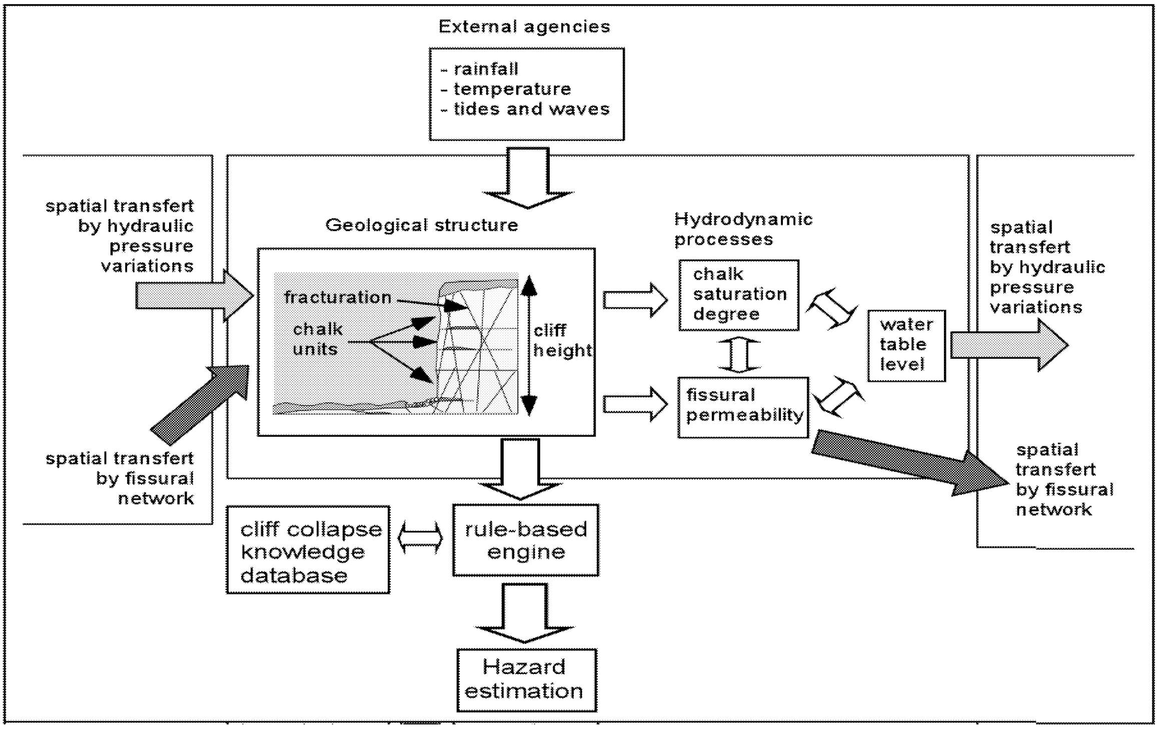
Mortimore R.N., Lawrence J., Pope D., Duperret A., Genter A., 2004, Coastal cliff geohazards in weak rocks : the UK chalk cliffs of Sussex. *in* : Mortimore R.N. and Duperret A. (eds) *Coastal chalk cliff instability*. Geological society, London, Enginneering Geology Special Publications, 20, 3-31.

ANNE DUPERRET is maitre de conférences in earth sciences at Le Havre University since 10 years. She is working on coastal erosion processes. She has a large experience on coastal chalk cliff instabilities, due to field works in collaboration with French Geological Survey (BRGM) and Brighton University (UK) on coastal chalk cliffs from France and United Kingdom.

CYRILLE BERTELLE is professor in computer science in Le Havre University and develops research activities in complex systems modeling. His current activities concerns self-organization processes formalization within various applications: ecosystems and natural systems, decision making modeling based on GIS models, emotional modeling for decision making. He is one of the co-directors of a research laboratories amalgamation which promotes the Sciences and Technologies in Information and Communication over the Haute-Normandie in France.

PIERRE LAVILLE is geologist at BRGM (French Geological Survey). He has been director of the regional geologic survey at Reims and is now responsible of the professionnal learning, the use and the management of BRGM geological database at Paris. He is a specialist of GIS system dedicated to earth sciences problems.

Figure 5 : conceptuel process modelling by multi-scale rule-based qualitative system



STRUCTURAL AND DYNAMIC COMPLEXITIES OF RISK AND CATASTROPHE SYSTEMS: AN APPROACH BY SYSTEM DYNAMICS MODELLING

Provitolo Damienne
Chargée de recherche CNRS
UMR 6049 ThéMA
Université de Franche-Comté
30 rue Mégevand
25030 Besançon Cédex
damienne.provitolo@univ-fcomte.fr

KEYWORDS
Dynamic modelling, General Systems Theory, Risk analysis.

ABSTRACT

Risk and catastrophe are complex systems. Within the scope of this paper, we focus our attention on structural and dynamic complexities of catastrophes and on the possibility of modelling and simulating its double complexity with a formal and methodological framework: the General Systems Theory and System Dynamics modelling. We then briefly propose a model of urban catastrophe related to a flood. We then propose some ways of research allowing exceeding the limits related to the modelling.

INTRODUCTION

This document aims to apprehend the complexity of the systems of risks and catastrophes in urban environments. The catastrophe is a social and spatial disorganization of the territorial system which is affected by a disturbing event. Based on scientific research on risks and disasters and on our own research, we could identify various forms of complexity. Some of them concern the structural complexity of catastrophes, while others are related to the complexity of spatial and temporal scales. Others still depend on the complexity of spatial forms and refer to the fractality of disasters. Finally, a last form of complexity is related to nonlinear dynamics. These various forms of complexity don't exclude each other, but could be combined. Within the scope of this paper, we focus our attention on structural and dynamic complexities of catastrophes and on the possibility of modelling and simulating its double complexity with a formal and methodological framework: the General Systems Theory and System Dynamics modelling. We then briefly propose a model of urban catastrophe related to a flood. We then propose some ways of research allowing exceeding the limits related to the modelling.

STRUCTURAL AND DYNAMIC COMPLEXITIES OF RISK AND CATASTROPHE SYSTEMS

Sciences of complexity propose a holistic approach to understand these phenomena. The holistic analysis seeks to understand systems mechanics by focusing not only on the entities which compose a system, but on the relations existing between these entities (Ménard et al. 2005). Initially we will try to identify structural complexity of risk and catastrophe systems by undertaking this holistic approach. Two levels of structural complexities could then be identified. The first level depends on the even definition of catastrophe. This is defined by risk specialists as a combination of hazard and vulnerability (R = A * V). However we prefer to abandon this term of combination, which doesn't integrate interactions between constituents and define disaster as a complex set of hazard(s) and vulnerability(ies). These two entities form the core constituents of catastrophe systems and their global functioning. In the absence of one or the other of these constituents, there cannot be a disaster. Thus, catastrophes are emergent phenomena of hazard and vulnerability. The emergence means that the global properties of the system cannot be deducted from the only knowledge of these entities (Zwirn 2005). If we neglect an essential aspect of a complex system, we cannot understand the system in its entirety. This means that the risk can't be reduced neither to the one or other one of its parts nor to the sum of its parts. This is an important finding. For a long time, the terms of risk and hazard were used as synonyms, especially in the field of natural risks, such as floods and earthquakes. As a consequence this simplified version of the concept of catastrophe had to favour reductionist approaches in this domain. Scientific research widely privileged the study of risks only from the angle of the hazard. Of course, scientific progress in this domain was particularly important. In fact, by neglecting the vulnerability entity and therefore omitting a part of the system, important aspects of the structure and the global behaviour of disasters were missed out. Initially, in the Anglo-Saxon literature, the term "hazard" is used to describe as well hazard as risk, two terms which are in fact very different. The sciences of the complexity thus developed the concept of risk.

The other type of complexity is relative to the increase of the levels of complexity when we go from some sector-related complex risk (hazard, vulnerability) to transversal complex risk (hazard, vulnerability and "domino effects") (Dauphiné 2003). This increase of the levels of complexity results not only from the even higher number of constituents but also and especially from the multiplicity of the interactions which unite the various entities of a catastrophe. So, the transversal complex risk integrates the varied nature of "domino effects": natural and technical or technological, natural and social or still, to take a last example, natural, technical and social. These "domino effects", particularly in urban areas, are creative of clearing of multiple thresholds of gravity in varied domains and for the same "hyperdisaster" (Guihou et al. 2006). This structural complexity is rarely taken into account by the managers of disasters, which often study only one category of disaster: the risk of floods, earthquakes, forest fires, tsunamis, nuclear accidents etc. are identified in the various documents of prevention and management of the major risks (Municipal Document of Information on Major Risks). But these various documents don't integrate the transverse complexity of catastrophes which take place in urban areas. Some exceptions exist in this domain, notably in Japan, where the authorities of Tokyo are afraid of a chain of natural and technological disasters following an earthquake (Hadfield 1992).

The Cartesian approach, which is limited mostly to the analysis of one type of risk, does not support the understanding of all the mechanics of a disaster. This scientific method tends at present to be replaced by holistic approaches. For the better understanding of a catastrophic event, the study of relations and interactions is better than the study of the system constituents.

The second form of complexity depends on the dynamic and unpredictable complexities of the disaster systems. The General System Theory of Ludwig von Bertalanffy and the systemic modelling offer a formal frame to build models of disasters which are centred on the complexity of the relations man / nature and on nonlinear dynamics. Within the framework of the General System Theory, the System Dynamic of J.W. Forrester (Forrester 1984; Aracil 1984) is interested in the changes which occur inside the studied systems. This method of modelling and simulation of complex systems appeared in the early 1960s when J.W. Forrester was a professor at "Sloan School of management" of MIT (Massachusetts Institute of Technology). They were developed in France from 1980 on, with the translation into French of his book "Principles of Systems". The models allowing apprehending the dynamics of a system base themselves on the concepts of interaction, feedback loops and complexity. The dynamic and unpredictable complexities of catastrophe systems result from feedback loops (a circle of cause-and-effect) which connect the different variables of the system, the interactions between these feedback loops and the delays between variables and nonlinear phenomena (Donnadieu and Karsky 2002). Two kinds of feedback relationships govern the dynamic systems of a catastrophe. Some push the system towards instability and disorder. These are positive feedback loops. Others make the system return to its initial state. These are negative feedback loops. These two kinds of feedback relationships operate in complex systems. The presence of these circuits of feedback loops, which can occur either simultaneously or successively, does not allow, without modelling and simulation, to predict the temporal dynamics of a system. This dynamic complexity will be increased by the presence of a temporal gap and non-linearity between variables (no proportionality between cause and effect). In the event of a disaster, the planning and the intervention of disaster managers are forces which, normally but not systematically, bring the system to stability and order. On the other hand, negative forces can push the system to disorder and to instability.

The System Dynamics modelling was thus retained to build models of urban disasters. The construction of a model starts by the realization of a Forrester diagram. This diagram represents the various elements which compose the system in term of stocks and flows, and clarify the relations established between the various variables of the modelled system. It is what we call the "Dynamo language". The relations between the variables of the system are mathematically formulated with for example statistical laws, logical rules (if then else). In this way quantitative and qualitative data can be integrated into the model. The basic elements of the model are the state variables, flows, converters and connectors. It is difficult to separate this approach of the computer tool allowing its application.

SYSTEM DYNAMICS MODELLING OF CATASTROPHE WITH THE STELLA RESEARCH PROGRAM

At the moment, Stella Research ® is one of the most wide-spread software to model and simulate the complex systems according to the principles of a formalization of Forrester. This software includes two modules: a graphic module which supports building the structure of the model. A mathematical module presents the results in the form of curves which is a set of differential equations defined by the graphic module. These differential equations are discretized in difference equations in the software Stella Research ®. The main symbols of the graphic module of the software Stella Research ® are represented in the table below:

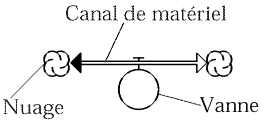
□	Stocks or reservoirs, the graphic formalism of it is a rectangle, are state variables. Their quantity varies through time according to the inflows and outflows. The value of these stocks informs about the state of the system all the time t. Stocks are used to represent so many material accumulations (the water, the individuals) as immaterial (the knowledge)
	Flows feed the reservoir and thus modify the state. They determine the variations at the various levels of the system. In the absence of flows, no change in the magnitude of stocks is possible. So, stocks and flows are inseparable.
○	The auxiliary variables or converters, represented by a circle, appear in connectors. They can be a constant, or a function according to time t or of a some variable a. These auxiliary variables are very useful to integrate qualitative information and delays into the models. They also allow to couple flows of different nature, for example flows of motorcars and motorist flows.
⊖——►	The connectors allow to link together the variables of the system and to simulate feedback loops. Connectors link stocks to converters, stocks to flow, converters to flow and converters to other converters.

Table 1: Presentation of the main symbols of the graphic module of the software Stella Research ®

This software was used to create a model of urban disaster of natural origin, a flood. This model (figure 1) has already been the object of publications (Provitolo 2003, 2005). We invite the reader interested in greater details of the structure and the results of the simulation to refer to these publications. The originality of this model is to associate various entities of disaster and its “domino effects”, namely the flood hazard (in this case, we have only considered river floods), the vulnerability of the population during the flood, the problems of urban traffic and of evacuation of the motorists as well as a module of panic. The system of disaster is in this case constituted by four modules in interaction. The module of hazard establishes a link between the modules of the vulnerability of the population subject to a flood. Indeed, the exposed population is a function of the area (km^2) flooded by river flood (module hazard). This hazard, physical phenomenon at the origin of the damages, is also going to have impacts on the urban traffic. It can provoke movements of panic amongst the motorists taken by the streams. These movements of panic, which can lead to acts of abandonment of vehicles, go then retroact on the module of the vulnerability by delaying the arrival of emergency services. This global model of disaster thus associates aspects generally treated independently of each other. Every module is constituted, under graphic shape, by state variables, flows and connectors. These graphic models are then transformed into computer models which allow running simulations. The results of simulations are curves which show the temporal dynamics of the catastrophe.

According to the values of the various parameters of the model (policies of prevention, rapidity of intervention of the emergency services, rate of contact and contagion of the panic), the forms of the dynamics of the catastrophe system modify. So, the system knows different evolutions. It does not adopt the same behaviour. Also the simulations lead to results which are counter-intuitive, unpredictable: the decrease of the rate of contact and contagion which build the relation between panic and not panic persons does not decrease the stocks of "contaminated" population, here, that of panic motorists. With nonlinear dynamics, the system can thus adopt unpredictable behaviour. These results of the simulation, by giving various possible evolutions of the system, are a tool of understanding. They also impose a rule of conduct. The models have to be the object of simulations based not only on average but also on extreme values of parameters. It is the most effective way of observing the variety of trajectories.

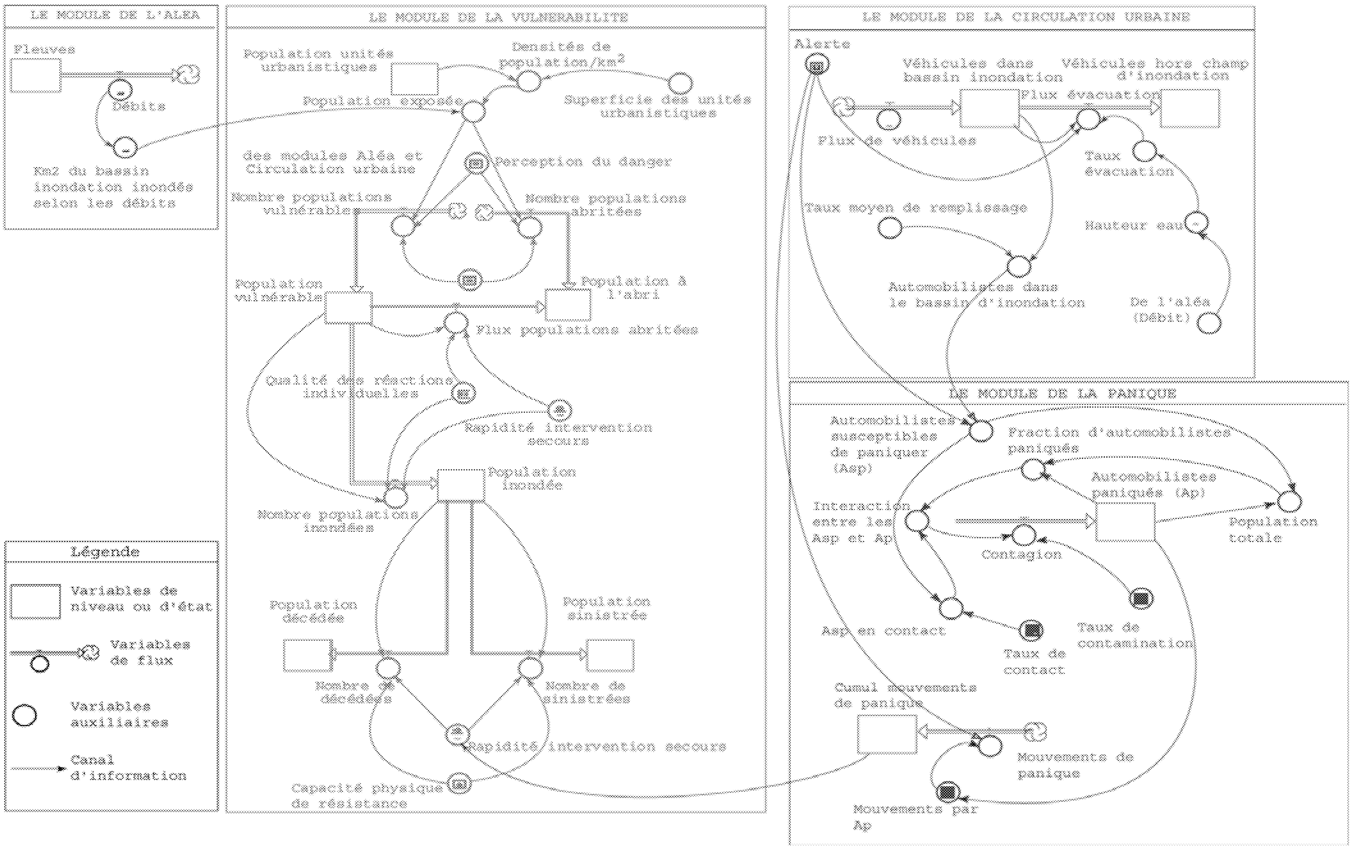

LE MODULE DE L'ALEA
Fleuves
Débits
Km2 du bassin inondation inondés selon les débits
LE MODULE DE LA VULNERABILITE
Population unités urbanistiques
Densités de population/km^2
Superficie des unités urbanistiques
Population exposée
des modules Aléa et Circulation urbaine
Perception du danger
Nombre populations vulnérables
Nombre populations abritées
Population vulnérable
Population à l'abri
Flux populations abritées
Qualité des réactions individuelles
Rapidité intervention secours
Population inondée
Nombre populations inondées
Population décédée
Population sinistrée
Nombre de décédées
Nombre de sinistrées
Rapidité intervention secours
Capacité physique de résistance
Légende
Variables de niveau ou d'état
Variables de flux
Variables auxiliaires
Canal d'information
LE MODULE DE LA CIRCULATION URBAINE
Alerte
Véhicules dans bassin inondation
Véhicules hors champ d'inondation
Flux évacuation
Flux de véhicules
Taux évacuation
Taux moyen de remplissage
Hauteur eau
Automobilistes dans le bassin d'inondation
De l'aléa (Débit)
LE MODULE DE LA PANIQUE
Automobilistes susceptibles de paniquer (Asp)
Fraction d'automobilistes paniqués
Automobilistes paniqués (Ap)
Interaction entre les Asp et Ap
Contagion
Population totale
Asp en contact
Taux de contact
Taux de contamination
Cumul mouvements de panique
Mouvements de panique
Mouvements par Ap

Figure 1: A catastrophe system

CONCLUSION

This System Dynamics modelling offers a formal and methodological frame to apprehend the complex structures of type of disasters (natural, technological, social risk) and different kinds (flood, earthquake, terrorism) as well as their temporal dynamics. This modelling allows analysing the global nature of catastrophes. However this approach has limitations, notably with respect to the consideration of the space. The space is indeed indirectly integrated into the models, for example by means of densities or of surfaces. To exceed this limitation, the "systemicien" can use other tools of modelling. The scientists have two solutions to integrate space in all its heterogeneousness and realize models which take into account temporal and spatial dynamics of a catastrophic event. The first solution is to link a Geographic Information System (G.I.S.) with a model of system dynamics to integrate the spatial constituent. Currently, the G.I.S. allows combinatorial analyses and structures, but on the other hand is less capable of modelling and simulating dynamics and complex dynamics. It is very likely that these limitations will be overcome within the next years. The second approach is to combine System Dynamics modelling with cellular automata. This research is in progress at the University of Maryland. The philosophy of this spatio-temporal modelling is to integrate the spatial systems, the simulation and the complexity into the problem of risk. So we could have the possibility of working on a spatial grid of square unit cells (raster) representing an urban area. In each of these cells a model of System Dynamics of catastrophe (model of type stock-flow) would be connected. The same structure of model would run in each cell. In this way we would obtain interactions between the cell and the systemic model, namely between the urban shape and the variables of the catastrophe system (hazard, vulnerability and "domino effects"). This architecture of models in interaction would allow understanding not only the spatial dynamics but also the temporal dynamics of the catastrophe. These research efforts are certainly worth to be further developed. Indeed, the knowledge obtained by these simulations would certainly allow taking up new forms of management of disasters.

REFERENCES

Aracil J. 1984. *Introduction à la dynamique des systèmes*. Presses Universitaires de Lyon, Lyon, 412 p.

Von Bertalanffy L. 1973. *Théorie générale des systèmes*. Dunod, Paris, 296 p.

Dauphiné A. 2003. *Risques et catastrophes*. Armand Colin, 2ème édition, Paris, 288 p.

Dauphiné A. 2003. *Les théories de la complexité chez les géographes*. Economica, Paris, 248 p.

Donnadieu G. and Karsky M. 2002. *La systémique, penser et agir dans la complexité*. Ed. Liaisons, Rueil-Malmaison, 269 p.

Forrester J.W. 1984. *Principes des systèmes*. Presses Universitaires de Lyon, Lyon.

Guihou X. ; Lagadec P. ; Lagadec E. 2006. « Les crises hors cadres et les grands réseaux vitaux- Katrina ». Mission de retour d'expérience, EDF, 34 p. http://www.patricklagadec.net/fr/pdf/EDF_Katrina_Rex_Faits_marquants.pdf

Hadfield P. 1992. *Tokyo séisme : 60 secondes qui vont changer le monde*. Ed. Autrement, Paris, 149 p.

Ménard A. ; Filotas E. ; J. Marceau D ; 2005. « Automates cellulaires et complexité : perspectives géographiques ». http://www.iag.asso.fr/aarticles/AUTOMATES

Provitolo D. 2005. « Un exemple d'effets de dominos : la panique dans les catastrophes urbaines ». *Cybergéo,* n° 328.

Provitolo D. 2003. « Modélisation et simulation de catastrophe urbaine : le couplage de l'aléa et de la vulnérabilité ». *Actes du colloque SIRNAT, 29-30 janvier 2003, La Prévention des Risques Naturels, Orléans,* http://www.brgm.fr/divers/sirnatActesColl.htm

Stella Research An Introduction to Systems Thinking. 1997. Hanover: High Performance Systems.

Zwirn H. 2005. « Qu'est ce que l'émergence ». Hors-Série *Sciences et Avenir*, juillet - août 2005.

COLLECTIVE INTELLIGENCE AND NEURAL LEARNING

MULTI OBJECTIVE OPTIMIZATION USING ANT COLONIES

Feïza GHEZAIL
LIMOS, IFMA, Institut Français
de Mécanique Avancée,
URAII, INSAT, Institut National
des Sciences Appliquées et de
Technologie

Henri PIERREVAL
LIMOS UMR CNRS 6158
IFMA, Institut Français de
Mécanique Avancée, Campus
des Cézeaux, BP 265, F-63175
Aubière Cedex, France

Sonia HAJRI-GABOUJ
URAII, INSAT, Institut National
des Sciences Appliquées et de
Technologie, Centre urbain nord
BP 676, 1080 Tunis

KEYWORDS
Multi objective, ant colony.

INTRODUCTION

Ant colonies are more and more used to solve various optimization problems, such as scheduling problems. In practice, it is often necessary to take into account several objectives in the optimization procedure. In this respect, ant colonies algorithms have to be adapted to be able to find a set of good solutions that cover in the best way the various regions of the Pareto front. In the following, we suggest an approach that can be used to address optimization problems with a few objectives. We will focus on visibility and desirability issues to favour diversity of solutions in the Pareto front. Further research direction will also be highlighted.

MULTI OBJECTIVE ANT COLONY OPTIMIZATION

Several articles related to multi objective ant colony optimization have already been published. Gravel *et al.*, 2002 address a multi objective scheduling problem. They use multiple visibility measures that they combine to determine the global visibily of an ant. The global update of the pheromone is based on the best solution found, at each ant cycle, using a function aggregating the three objectives handled. A sequencing problem is presented by McMullen, 2001. Two objectives are considered: to minimize setups and stability of material usage rate. Only one visibility measure is used; the pheromone is updated according to the smallest Euclidean distance computed. Doerner *et al.*, 2006describe a multi objective project portfolio selection problem. The update of the pheromone trail is based on the two bests solutions obtained at each run for each objective handled. A Pareto archive is used to store the non dominated solutions. A reliability optimization problem is addressed by Zhao *et al.* (still in press). The two visibility measures are reduced to a single one using a ratio. Pinto and Barãn, 2005 solved a multicast routing problem using two different algorithms: a Multi-objective Ant Colony Optimization Algorithm and a Multi objective Min-Max Ant System. A Pareto archive is used to update the pheromone trail. However, methods that would aim at favouring diversity of solutions in the Pareto set are not described. In the approach presented next, emphasis is put on searching for this diversity.

MULTI OBJECTIVE ANT COLONY APPROACH

1. General framework

Dealing with several objectives in ant colonies that use principles proposed by Dorigo and Gambadella, 1997, necessitates to answer three questions: (1) how to globally update pheromone according to the performance of each solution on each objective, (2) how does a given ant locally selects a path, according to the visibility and the desirability, at a given step of the algorithm (3) how to build the Pareto front. Figure 1, summarized the main steps of such an algorithm.

Step 1 Initialize the pheromone trail and initialize the Pareto set to an empty set

Step 2 For each ant, compute the visibility measures associated with each objective, so as to select the successive nodes according to visibility and pheromone amount, and locally update the pheromone trail until all nodes selected

Step 3 Try to improve the obtained solutions using a local search

Step 4 Evaluate the obtained solutions according to the different objectives and update the Pareto archive with the non dominated ones and reduce the size of the archive if necessary

Step 5 Identify several best solutions according to the different objectives considered

Step 6 Globally update the pheromone, according to the best solutions computed at step 5
Iterate from Step 2 until the maximum of iterations is reached.

Figure 1: General procedure of the proposed algorithm

2. Pheromone and desirability

Each time ants select successive paths to construct a solution, the pheromone trail τ_{ij} of each segments (i,j), cant be locally updated according to:

$$\tau_{ij}(t+1) = (1-\rho).\tau_{ij}(t) + \rho.\tau_0 \qquad (1)$$

$\rho\,(0 < \rho < 1)$ is a persistence factor and τ_0 is a constant. This local update of the pheromone is used to evaporate some quantitiy of pheromone to avoid a premature convergence of the algorithm.

Then, at the end of an iteration, every ant has found a solution s and the pheromone trails have to be globally updated on the basis of the performance achieved on the u objectives f_1, …f_u. To select the set of paths for which pheromone has to be reinforced, we determine, from the set of available solutions, those that have yield the best results on w linear combinations of the objectives:

$$F^p(s) = \lambda_1^p \cdot f_1(s) + \lambda_2^p \cdot f_2(s) + ... + \lambda_u^p . f_u(s) \ , \ p = 1,...,w \ .$$

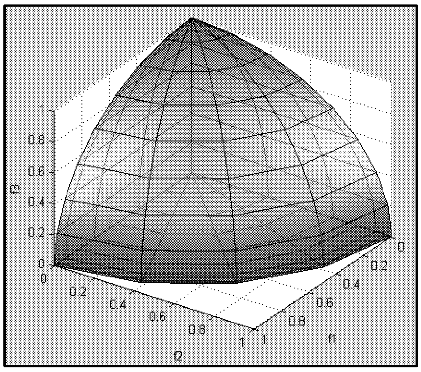


Figure 2: Search directions for a maximization problem

These functions characterize w search directions, which can be determined through the w vectors: $(\lambda_1^1,...,\lambda_u^1),(\lambda_1^2,...,\lambda_u^2),...,(\lambda_1^p,...,\lambda_u^p),...,(\lambda_1^w,...,\lambda_u^w)$ as illustrated in Figure 2 in the case of three objectives. This incites the algorithm to explore systematically distinct areas, so as to favor the diversity of solutions in the Pareto set (Siarry and Collette, 2002). Let s_{best}^p be the solution that yields the best results with F^p. Then the pheromone of each of the n-1 segments (i,j) of the corresponding path is reinforced in a minimization problem as follows:

$$\tau_{ij}(t+n) = (1-\rho).\tau_{ij}(t) + \rho.\Delta\tau_{ij}^p \ , \ p = 1,...,w \qquad (1)$$

$$\Delta\tau_{ij}^p = \begin{cases} \dfrac{1}{F^p(s_{best}^{\text{p}})} & \text{, if (i, j) } \in \text{best solution according to the u - ple p} \\ 0 & \text{otherwise} \qquad (2) \end{cases}$$

Where $\rho\,(0 < \rho < 1)$ is a persistence factor, t the current discrete time and n the number of nodes of a path. Let us note that this approach is adapted if the number u of objective is low.

3. Visibility

In addition to the pheromone quantity, ants are guided by a proximity measure called *visibility*. Since several objectives are considered (Liao and Juan, 2006), several visibility measures can be used, depending on the problem. Visibility values can be stored in a matrix connecting each node i to each node j. For example, in Gravel *et al.*, 2002, a visibility measure $\eta_{\text{ij}}^{\text{c}}$ is defined for each objective c (c = 1, …, u) and combined. Then, each ant k (k = 1, …, m) that leaves node i selects the next node j to be visited according to the probability given in (3), where q is a randomly generated variable and q_0 is a parameter, such that $q \geq 0$, $q_0 \leq 1$. α and β_c are the control parameters and tabu$_\text{k}$ is a memory list used to avoid reselection of nodes already chosen by each ant k.

$$p_{ij}^{\ k} = \begin{cases} \dfrac{[\tau_{ij}(\text{t})]^\alpha \cdot \prod\limits_{c=1,...u} \left[\dfrac{1}{\eta_{\text{ij}}^{\text{c}}} \right]^{\beta_c}}{\sum\limits_{l \notin tabu_k} [\tau_{il}(\text{t})]^\alpha \cdot \prod\limits_{c=1,...u} \left[\dfrac{1}{\eta_{\text{il}}^{\text{c}}} \right]^{\beta_c}} & \text{if j} \notin \text{tabu}_\text{k} \\ 0 \quad \text{otherwise} \end{cases} \qquad (3)$$

4. Multi objective Local improvment

According to Hu et al. (2005), an important weakness of the Ant Colony algorithm is that the search may fall into a local optimum. An improvement function, multi objective in our case is useful to enhance the ACO performance. A possible approach consists in selecting p solutions from those generated by the algorithm and in modifying them with some elementary mofifications (e.g. 2-OPT for a scheduling problem). Then, the best l ones are stored in the Pareto archive. To select the l best solutions, l directions are defined to favor the less populated area of the current Pareto front, so as to improve the diversity of the proposed solutions to the decision maker.

5. Pareto selection

The set of non-dominated solutions is stored in an archive. During the optimization search, this set, which represents the Pareto front, is updated (Loukil *et al.*, 2005). At each iteration, the current solutions obtained are compared to those stored in the Pareto archive; the dominated ones are removed and the non dominated ones are added to the set. The size of this set needs to be kept reasonable, which may imply to sometimes remove non dominated solutions. As suggested for multi objective genetic algorithms, to preserve the diversity of the set, solutions belonging to the most populated areas can be removed first.

EXAMPLE

We tested the proposed approach for a single machine multi objective problem related to a printing shop. Each product has a size and a printing label that needs different ink colours, which induce constraints about the tool to be used (mandrel) and on the sequence of jobs. We consider groups of jobs having the same size to be scheduled. We are interested in minimizing a performance function (based on ink changes and total tardiness), a robustness measure (based on a regret in case of machine breakdown), and a flexibility measure that quantifies possible lost of performance if the tool was not available (see for more details Ghezail *et al.*, 2005). At each iteration of the algorithm, the number of solutions in the Pareto set staid low, so there was no need to eliminate solutions in the archive. The resulting Pareto set allows the decision maker to select the most suited schedule, according to the operating conditions of the workshop.

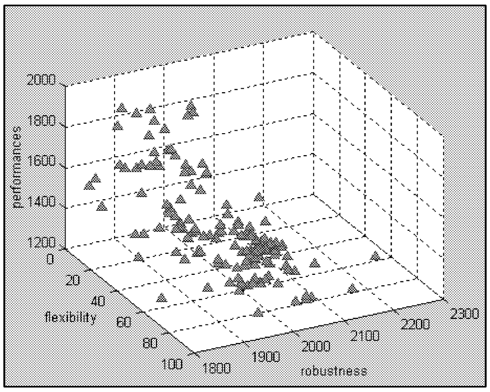


Figure 3: The Pareto set distribution

CONCLUSION

The method proposed here aims at improving the diversity in the Pareto set, so as to offer different types of compromises to the decision makers. The key principles are simple to implement in an ant algorithm, but they are mainly suited for problems with few objectives. One remaining important research issue is related to the adaptation of visibility principles to multi objective problems. Currently, the different visibility measures are aggregated in a single one using weights or a ratio. An interesting research direction would consist in adapting the influence of each visibility measures depending on what solutions are available in the Pareto set.

REFERENCES

Doerner, K.F., Gutjahr, W.J., Hartl, R.F., Strauss, C., Stummer, C. 2006. "Pareto ant colony optimization with ILP preprocessing in multiobjective project portfolio selection." European Journal of Operational Research, **171**, 830-841.

Dorigo, M., and Gambadella, L. M. 1997. "Ant colonies for the travelling salesman problem." BioSystems, **43**, 73-81.

Dréo, J. 2003. « Adaptation de la méthode des colonies de fourmis pour l'optimisation en variables continues. Application en génie biomédical. » PhD Dissertation, University of Paris 12, France.

Ghezail, F., Hajri-Gabouj, S., and Pierreval, H. 2005. "A multiobjective Ant Colony Approach for a Robust Single Machine Scheduling problem." *Proc. of International Conference on Industrial Engineering and System Management, Marrakech, Maroc*, 479-488.

Gravel, M., Gagné, C., and Price, W. L. 2002. « Algorithme d'optimisation par colonies de fourmis avec matrices de visibilité multiples pour la résolution d'un problème d'ordonnancement industriel. » *INFOR*, **40**(3), 259-276.

Hu, Y-H., Yan, J-Q., YE, F-F., Yu, J-H. 2005. « Flow shop rescheduling problem under rush orders." Journal of Zhejiang University SCIENCE **6A**(10), 1040-1046.

Liao, C-J., and Juan, H-C. (2006). "Ant Colony optimization for single machine tardiness scheduling with sequence-dependent setups." Computers and Operation Research, in press.

Loukil, T., Teghem, J., and Tuyttens, D. 2005. "Solving Multi-objective Production Scheduling using metaheuristics." European Journal of Operational Research, **161**, 42-61.

McMullen, P.R. 2001. "An ant colony optimization approach to addressing a JIT sequencing problem with multiple objectives." Artificial Intelligence in Engineering, **15**, 309-317.

Pinto, D., Barãn, B. 2005. "Solving Multiobjective Multicast Routing Problem with a new Ant Colony Optimization approach." *In proceeding of IFIP/ACM Latin-American Networking Conference LANC'O5, Cali, Colombia.*

Siarry, P., and Collette, Y. 2002. "Optimisation multiobjectif. » Eyrolles, Paris, France.

Zhao, J.H., Liu, Z., Dao, M.T. (in press). "Reliability optimization using multiobjective ant colony system approaches." Reliability Engineering and System Safety.

SELF-ORGANIZATION IN AN ARTIFICIAL IMMUNE NETWORK SYSTEM

Julien Franzolini
LITIS - University of Le Havre
25 rue Ph. Leblon - BP 540
76058 Le Havre Cedex - France
julien.franzolini@wanadoo.fr

Damien Olivier
LITIS - University of Le Havre
25 rue Ph. Leblon - BP 540
76058 Le Havre Cedex - France
damien.olivier@univ-lehavre.fr

KEYWORDS

AIS, immune network, idiotypic network, self-organisation.

ABSTRACT

Artificial Immune System field uses of the natural immune system as a metaphor for computational problems. The immune system exhibits a highly distributed, adaptive and self-organizing behavior. Furthermore it can learn to recognize shapes with its adaptive memory. The approach explored is inspired by an immune network model like Stewart-Varela's model. This model is designed to better understand the memory and cognition properties. Emergent antibodies configurations are studied on this immune network model; indeed these configurations seem like cellular automata configurations and appear with self-organizing properties.

INTRODUCTION

The immune system is a natural complex system which protects all vertebral organisms. It is able to recognize foreign molecules and to learn to detect this foreign agent more quickly and more effectively. This system has many proprieties: it is adaptive, self-organizing, distributed and robust. All these capabilities interest the computer sciences for various domains : clustering, intrusion detection and optimization. This paper explains fundamental biological theories used in artificial immune systems and in a second time it presents a model of immune system based on immune network theory.

THE IMMUNE SYSTEM

General description of the immune system

The immune system is really a complex system due to the number of actors and the multiple interactions involved in an immunizing response. This system protects the body against pathogen agents by various collective mechanisms. These mechanisms termed immunity give the state of protection against a foreign agent called antigen. The immune system is composed of lymphocytes which are known under the name white blood cells, more exactly B and T cells. These B cells help the process of antigens recognition by secreting antibodies corresponding to an antigen. Antibodies can fix antigens by a complementary shape (figure 1).

Antigens are majors actors because they are the attractor of the immunizing response, without these antigens the system does not engage a response. There is two type of immune response:

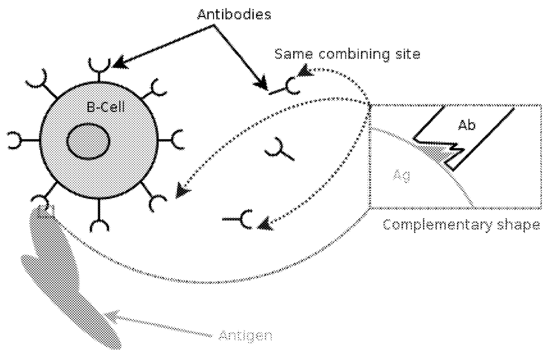


Figure 1: Affinity Between a B Lymphocyte, its Antibodies and an Antigen.

- The primary response comes from the natural or innate immunity, this response is provoked when an antigen is encountered for the first time. The B lymphocyte specific to the antigen generates many antibodies oriented to the antigen shape. With this first action the antigens can be more easily destroyed.
- The secondary response called acquired or specific is occurred after a similar re-infection and not compulsory with an identical shape. The system can yet produce more quickly and more massively specific antibodies by memorizing the original shape of the past encounter [Jon Timmis, 2001]. This memorization of antigens is the occasion of an immunological debate to explain this adaptive memory .

Many models try to explain how the immune system reacts but theories are in contradiction, as the self-nonself theory and danger theory and some other theories are complementary as the clonal selection theory with the others then it is necessary to give a brief review of these different theories.

Clonal selection theory

When an immune response is engaged, lymphocytes are stimulated to proliferate and secrete their free antibodies corresponding to the antigen. During this proliferation which is realized with cloning, some mutations are made on the new B-cells. These mutations called hyper mutations somatic improve the capacity to fix the pointed antigen. A B-cell and its antigen are specific and code for the same shape. More the affinity of an antibody and an antigen is important more the B-cell is cloned. This phenomenon is known as the maturation of immune response [Leandro Nunes de Castro, 2000] and it is one of immune learning mechanism. The cell with high stimulation (high antigens affinity) creates with cloning to high living cells called memory cells, which are kept in the body during a long time to generate a secondary immune response. Such a minority of B lymphocytes can recognize one specific antigen and is activated by clonal selection then this proliferation increases the specific answer. This principle is nearly of natural selection used in evolutionary algorithm but it does not explain all memory mechanisms and still less how the system recognizes the foreign agents.

Self and nonself discrimination

The immune system can give an immune response and improves it by the clonal expansion but how the system doesn't recognize self antigen (or self agent)? This question is the beginning of an enlivened discussion between immunologists, and involves several interaction of lymphocytes. But the most used immunological explication is the *thymic negative selection of T-cells* [Leandro Nunes de Castro, 2003]. These T lymphocytes which finish their maturation in thymus became mature and are introduced in blood if they don't have affinity with self antigen. With this action, the domain recognition of T-cells maps the non-self universe. The system does not recognize non-self molecules as water, food, stomach bacteria... But these molecules or these organisms are foreign to the body. This problem gives an alternate hypothesis: Danger Theory.

Danger theory

The system has the ability to respond to foreign agent but only pathogenic and not only non-self. The danger theory explain that the system recognizes only dangerous invaders. This response is induced by a danger signal (ejection of molecules), when a cell is in stress or killed. In this theory, there are many danger signals not only in secreting molecules, and this different type drives the immune response [Uwe Aickelin, 2003]. This theory is a new way to use immune system for biological metaphor in particularly for intrusion detection system. All these variant theories are insufficient to explain the antibodies coverage of the diversity of dangerous shapes.

Immune network theory: idiotypic network

Over all this theory and hypothesis, the most important for this work is the immune network theory, which tries to explain the immune organization of the antibodies distribution. This theory is based on a theory established by Niels Jerne (Nobel Price for his work) that the lymphocytes are able to enter in interactions. This theory suggests that this interactions network can grow up unless antigens stimulation, to make the antibodies distribution. This theory reposes on a first hypothesis gived by Coutinho that the immune repertory is complete to be able to recognize all various antigens. This hypothesis was completed by Jerne: if the immune repertory is complete, antibodies of the same body have to enter in interaction by their complementary combining site, and so an antibody can provoke an immune response against an other antibody.

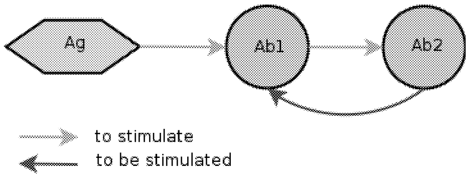


Figure 2: Ab/Ag and Ab/Ab Stimulations [Leandro Nunes de Castro, 2003]

The two hypotheses give the theory of a development in network called the idiotypic network. It shows how the system can have sufficient diversity to recognize unknown antigen. This stimulation network (figure 2) maps the universe of possible shapes.

AN IMMUNE NETWORK MODEL

Generally, the interaction from idiotypic network is simulated with the Farmer et Al. [Jon Timmis, 2002] equation which describes the antibody interaction between other antibodies, antigens and formulate its death rate :

$$Stimul = c[(antibodiesrecognize) - (I'mrecognized) \\ +(antigenrecognize)] - (death)$$

A development in networks is chosen, cause of its different properties as self-organization. This work relies on a research from Stewart and Varela [Stewart, 1994] where they show cognitive properties from an idiotypic network model. They have tried to make a mathematic model but it was highly non-linear and so not workable. They create so a computer model, where the idiotypic network is symbolised on a form space. In a form space, one dimension corresponds to one stereochemic characteristic, which describes the combining site. De facto, the distance of two entities in this space represents their characteristics differences (figure 3).

Two complementary molecules (Ag/Ab or Ab/Ab) able to fix each other are close in the space. In their model, antibodies were recruited if their stimulation (depend on affinity whit other complimentary antibodies) belongs to a recruitment window.

General principle of the conceived model

The Stewart-Varela affinity calculation between two antibodies is kept in this new model (equation 1), but the calculation of the stimulation received by an antibody is different (equation 2). Not only two types of complementary antibodies regrouped by plan are considered but n types so n plans. The stimulation undergoes by an antibody is the sum of all antibodies affinities of all other complemantary plans.

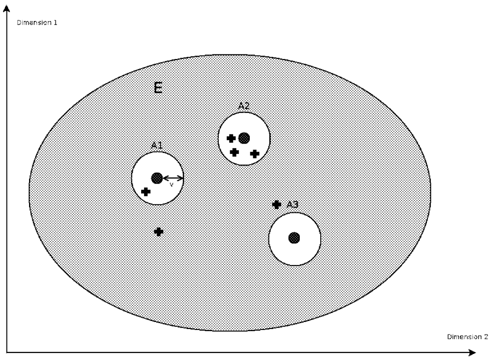


Figure 3: Two Dimensions Form Space.

$$m_{ij} = e^{-d_{ij}^2} \tag{1}$$

m_{ij} is the affinity and d_{ij} the distance between two antibodies i and j in the form space. An Euclidian distance is used but for example an Manhattan distance could be considered.

For a given plan p:

$$h_i = \sum_{k=1}^{k<nbPlan, k\neq p} \sum_{j=1}^{j<N_k} m_{ij} \tag{2}$$

h_i is the stimulation received by an antibody i belonging to the plan p, $nbPlan$ is the number of plans and N_k the antibodies number for a plan k.

The calculation of the distance is realized in a form space which has the topology of a tore. The simulation begins by an antibody introduction from a random plan in the center of the space. After, antibodies are randomly subjected in the recruitment to be introduced into the system. If one corresponds (its stimulation belongs to the window of recruitment $h_i \, \epsilon \,]b_{min}; b_{max}[$) the search for a candidate stops and the found antibody is introduced into the system.

The goal of this model is to develop itself in self-organized criticality, to maintain the criticality after each introduction, all antibodies stimulations are recalculate to be sure that all antibodies belong to the recruitment window. An antibody is rejected from the system if its stimulation is not sufficient or too high.

Details model

The details of the principal phase called development are given by the algorithm 1. This algorithm ends if the system becomes stable. The stability is defined by the fact that no antibody can be recruited in the system : the stimulation of all the possibilities of shapes doesn't agree with the recruitment window and in the previous phase all the stimulations were verified, so the system does not evolve any more.

In these algorithms, antibodies and lymphocytes are the same entities because they code for the same characteristics (attributes) and in this model there is no clonal expansion. The lymphocytes concentration for a point of the form space is 0 or 1.

Problems of collapse and death of systems can be encored. This fact comes from the suppression of all antibodies which cannot be maintained, it is the reason for the deletion of a unique antibody, particularly the one which undergoes most pressure (having the highest stimulation).

```
begin
    Introduction of a central lymphocyte ;
    Evolution ← true ;
    while Evolution do
        Evolution ← recruitmentLymphocyte() ;
        Calculate lymphocyte interactions in the
        different plans ;
        Deletion of a unique lymphocyte which don't
        agree with the recruitment window ;
        while deletion of one lymphocyte do
            Calculate lymphocyte interactions in the
            different plans ;
            Deletion of a unique lymphocyte which
            don't accord with the recruitment window ;
        end
    end
end
```

Algorithm 1: Principle of Development

To give freedom to the system, all plans and positions in the form space are randomly selected. The critical state of the system is maintained by continuing the elimination of the lymphocytes as long as their stimulations do not agree. This phenomenon is called distribution of avalanches.

```
while candidate exists do
    Select at random of a Lymphocyte among the
    population of candidate ;
    Select at random of a plan ;
    while plan opposed exists do
        Calculation of the lymphocyte interactions with
        each opposed plans;
        if can be recruit then
            insertion of antibody in the system;
            return true ;
        end
        Select at random of a new opposed plan ;
    end
    Select at random of a new candidate ;
end
return false ;
```

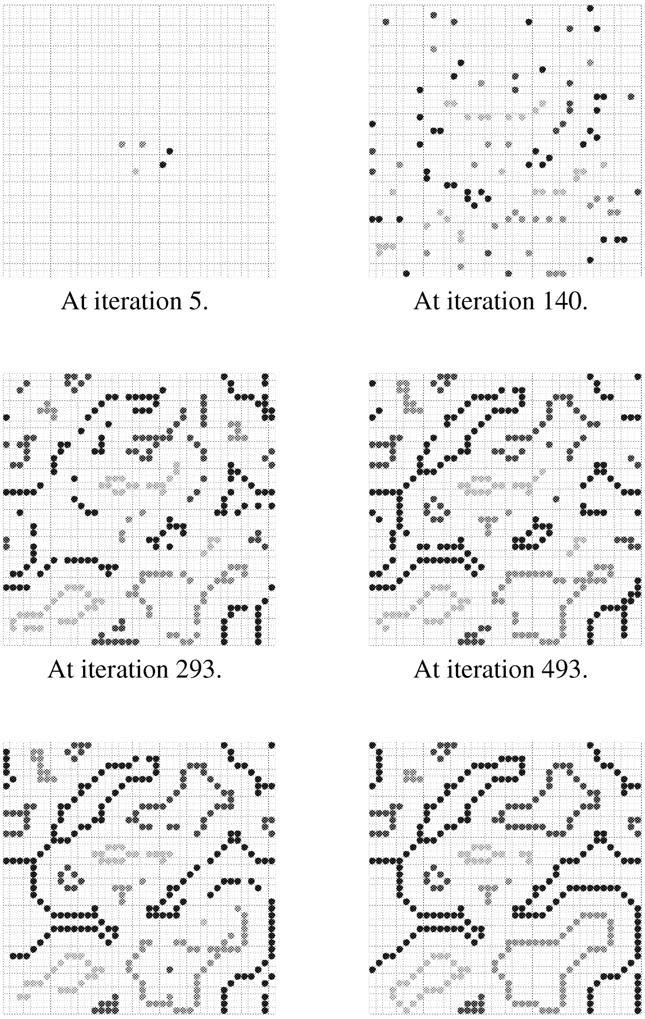
Algorithm 2: Recruitement Mechanism

RESULTS AND MODEL PROPERTIES

A self organizing system

Before running the system, it is necessary to choose adequate bounds. With arbitrary bounds, no developments emerge and on the contrary the lymphocytes fill the form space randomly without organization.

Figure 4 shows a development with self-organization in two dimensionnal form space (better for visualization). This form space has a size of 40 possibilities by dimension and the topology of the space is a tore. Each color represents a complementary plan. The system is started with the two bounds: $b_{min} = 2.10^{-10}$ and $b_{max} = 22.10^{-8}$; and with four complementary plans. At the beginning the system evolves from the first central lymphocyte and then enters in a organization phase to become stable at the iteration 898.



At iteration 5.

At iteration 140.

At iteration 293.

At iteration 493.

At iteration 672.

At iteration 898.

Figure 4: An Immune Network Development

Figure 5 gives the evolution of lymphocytes by plan and the global population for the previous example. For an other run, the obtained system has the same morphology but it has not an identical lymphocytes configuration due to the random submission to the recruitement. An iteration corresponds to a whole development phase, this phase contains a single introduction but no or many deletions. For each iteration the system has a critical configuration (All stimulations belong to the window).

A behavior dependent on bounds

The system can develop or not according to the bounds, and have a self-organized behavior or not. That seems like cellular automata where the behavior depends on the complexity class [Wolfram, 1983]. One of major results is that the bounds condition the development and the characteristics of the system: lymphocytes activities (recruitement or death), convergence time to a stable state or no stabilization and morphologies of emergent lymphocytes organization.

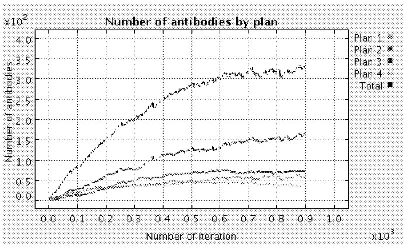


Figure 5: Lymphocyte Evolution by Plan during the Development

The example (figure 6) exhibits tree different systems with the same recruitement window: $b_{min} = 36.10^{-9}$, $b_{max} = 36.10^{-8}$. But the first has 2 plans and a size of 50, the second has 2 plans and a size of 100 and the last has 5 plans for a size of 50. The size and the number of plans have no influences on the system behavior.

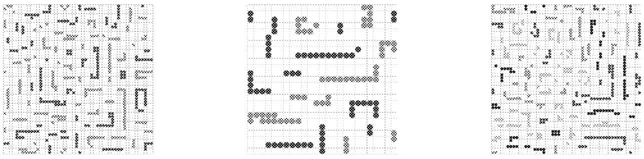


Figure 6: Identical Morphologies with the Same Bounds.

Self-organized criticality

One of caracteristics of the self-organized criticality (SOC) is the macroscopic behaviour. The self-organized critical systems displays a spatial scale-invariance (or time-invariance) characteristic of the critical point of a phase transition, but, unlike the latter, in SOC these features result without needing to tune control parameters to precise values [Bak, 1996].

The system exhibits auto-organized behaviors as the lymphocytes avalanches (lymphocytes deletions). To study this phenomenon, avalanches are ploted in double logarithm scales functions of the avlanches size and the number of occurences by size. The result (figure 8) for $b_{min} = 2.10^{-8}$ and $b_{max} = 2.10^{-7}$ is not a power law which is the most frequent scaling laws that describe the scale invariance found in many natural phenomena, but anyway a distribution.

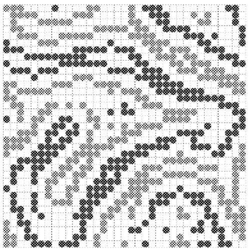


Figure 7: Unstable System with Lymphocytes Avalanches.

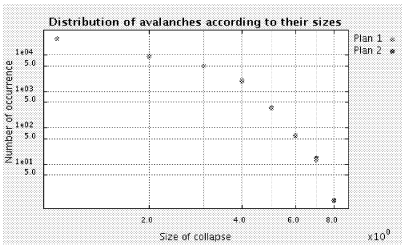


Figure 8: Avalanches Distribution Functions of Size.

Antigens reactions

To test the antigens reactions, a form is introduced in the system by forcing. This antigen does not undergo the phase of maintain or recruitement. An antigen is introduced figure 9 (square shape on the figure). This introduction changes the stimulations of the other lymphocytes and modifies the lymphocytes configuration.

Another aspect is that more the exposure of the antigen is long more the system is modified and conversely a little exposure does not distort the configuration. The exposure duration is corresponding to the number of iterations. With the theory [Stewart, 1994] according to the network morphology conditions the memory, it is supposed that the time of exposure amplifies or not the learning of the immune network by deformation of the spatial configuration.

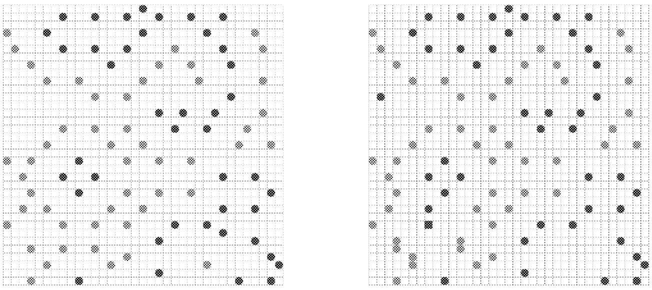


Figure 9: Antigen Introduction.

Comportements maps

One of this model problems is how to have bounds which would give self-organized properties, to find more easily valid bounds (conceivable development). Comportements maps are drawed by making vary bounds. These maps represent the number of iterations to converge for a death, a stable state or a no stabilization.

The blue color (figure 10) indicates the system death (system whitout lymphocytes), the green one represents a system still alive or stable and the black shows the impossible developments. A map is built for a given size and a given number of plans. The luminous intensity of the blue or the green indicates the speed of convergence to arrive at a given behavior, indeed at every iteration a test is realized to know if the system is empty or not, if it is stable or in development. An average on several launches with identical bounds is realized to give an usable map.

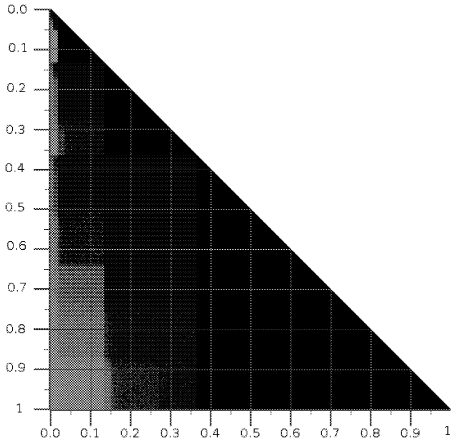


Figure 10: An Comportements Map with $b_{min}, b_{max} \in [0, 1]$, 2 Plans and a Size of 25 by Plan.

CONCLUSION

This model allows to approach better the different interactions which form the idiotypic network and so to understand how these very simple and multiple interactions can generate a memory and realize a learning.

One of the perspectives is to manage the behavior by bounds (maybe with comportements maps) to realize an artificial memory able to develop in self-organization and with a critical state to be adaptative.

REFERENCES

[Bak, 1996] Bak, P. (1996). *How Nature works : the science of self-oganized criticality.*

[Jon Timmis, 2001] Jon Timmis, Mark Neal, J. H. (2001). An artificial immune system for data analysis.

[Jon Timmis, 2002] Jon Timmis, T. K. (2002). *Artificial Immune Systems*, chapter 11, page 209. Idea Group Publishing.

[Leandro Nunes de Castro, 2000] Leandro Nunes de Castro, F. J. V. Z. (2000). An evolutionary immune network for data clustering.

[Leandro Nunes de Castro, 2003] Leandro Nunes de Castro, J. T. (2003). Artificial immune systems as a novel soft computing paradigm. *Soft Computing*, pages 526–544.

[Stewart, 1994] Stewart, J. (1994). Un système cognitif sans nerones: les capacité d'adaptation, d'apprentisage et de mémoire du système immunitaire. *Intellectica.*

[Uwe Aickelin, 2003] Uwe Aickelin, Peter Bentley, S. C. J. K. (2003). Danger theory: The link between ais and ids?

[Wolfram, 1983] Wolfram, S. (1983). Statistical mechanics of cellular automata. *Review of Modern Physics*, (55):601–644.

PYOCYANIC BACILLUS PROPAGATION SIMULATION

Antoine Dutot[1] Pierre Magal[2] Damien Olivier[1] Guilhelm Savin[1]
[1]LITIS, [2]LMAH
Université du Havre
UFR Sciences et Techniques
25 rue Philippe Lebon - BP 540
76058 Le Havre Cedex - France
firstname.lastname@univ-lehavre.fr

KEYWORDS

Pyocyanic bacillus, Antibiotic, Nosocomial, IBM, Ant Algorithm.

ABSTRACT

Nosocomial diseases are pathologies that appear during medical care that were not present at patient admission. Being able to simulate the propagation of such diseases inside an hospital and to track them is therefore important to fight and avoid them. The Pyocyanic Bacillus is a frequent example of such a disease. It is important to understand how such bacteria propagate since more and more of their strains become antibiotic resistant. Therefore we must not only be able to treat them, but also to block their diffusion to avoid them. The work presented here consist in a simulation of the propagation of pyos inside an hospital taking spatial problems into account, and allowing to better understand the infection diffusion mechanisms and to propose some means to circumvent it. The simulation considers both the spacial representation of the hospital, and the different actors, healthcare workers, patients and visitors.

INTRODUCTION

Pyocyanic Bacillus (Pyo) or Pseudomonas Aeruginosa (Figure 1) is a pathogen bacteria that is often found in eye, lung, or bladder infection and wound or burn infection for example. Pyos generate the pyogenic process, that is, pus formation. Indeed, bacterias multiply on the infection site, causing the massive production of neutrophils that phagocytose bacterias. These neutrophils die therefore producing pus.

Pyos are one of the most often encountered nosocomial disease. They are easily communicated by contact between patients and healthcare workers. Furthermore, Pseudomonas Aeruginosa is a bacteria that is naturally resitant to antibiotics and quickly adapt to medicine treatments. When antibiotic treatments have already been administered, this bacteria becomes particularly resistant, which is the case in hospitals. If such strains persist in an hospital they can severely complicate patient treatment.

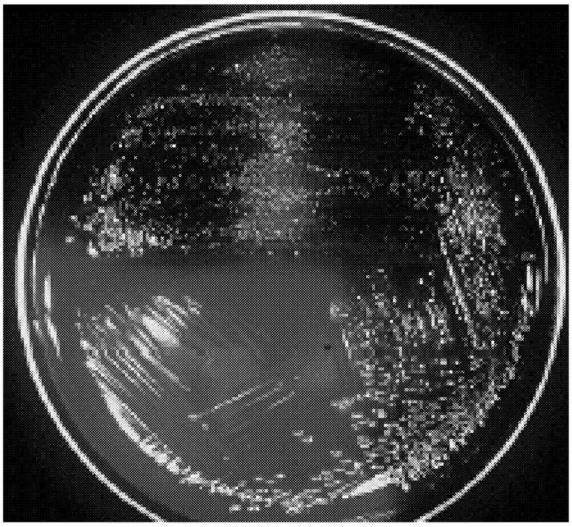


Figure 1: Pyocyanic Bacillus Colony

In [Giamarellou, 2002], Helen Giamarellou says that: “Almost 50 years ago, Pseudomonas aeruginosa was rarely considered as a real pathogen [...] the emerging resistance problem of Pseudomonas will worsen while the approaching era of ’the end of the antipseudomonal antibiotics’ will become a nosocomial nightmare.

Therefore, not only we must search means to treat it, but it is necessary to block its proliferation and communication ways.

We propose a simulation of the infection propagation using a model representing spatially an hospital, and using a concrete representation of individuals. Such a model is called an individual based model (IBM). This simulation should help us to track the infection diffusion and to discover its mechanisms, for example infection reservoirs, or the common passage points for the infection.

The model should then be used to both visualize the infection path from already collected data, and to simulate an hypothetical infection. With such a model, it is possible to develop algorithms that can detect infections paths and areas inside the hospital. Further this model could be used to devise a analytical and deterministic model whose trajectory study would be easier.

The next section details the model, and take as example the Jacques Monod Hospital from which series of data on the Pyocyanic Bacillus has been collected. In the following section, the simulation of an infection is described and preliminary results are presented. Finally, an algorithm allowing the detection of infected areas using the ant metaphor is discussed.

MODEL AND VISUALIZATION

The model focuses on a spacial representation of the hospital and, for the simulation, of individuals. We can distinguish at least two important classes of individuals : employees and patients. These classes can then be subdivided into sub-classes, for example doctors and nurses for employees.

To define a common spacial representation that could fit almost any kind of hospital, the building is represented as an undirected graph $G = (V, E)$ with V the set of vertices and E the set of edges. Vertices of the graph represent distinct sectors in the hospital, patient rooms, waiting-room, operating-room, etc. Edges of the graph represent paths between these sectors. The figure 2 shows the Jacques Monod hospital using this representation. On top the graph representing the hospital, under, the same graph but constrained by the physical layout of the building. This construction is arranged as a cross, making it easy to put a wing in quarantine, but also forcing the employees and visitors to use very common passage points.

Individuals move inside the graph from vertex to vertex following the edges, and interact one with another. Individuals have a path inside the graph. These path can be of two forms, they can be:

- predefined, if we are replaying data provided by an hospital;

- programmed, if we are simulating an infection.

For predefined paths, collected data often contains only series of locations or merely the starting sector and end sector. In this case the simulator is able to reconstruct a path following constraints. Often a A^* or Dijkstra algorithm are used to recreate the path. However using such algorithms can lead to different results since patient and healthcare worker displacements are often not optimized this way. Individuals can wait on each node, and cross edges at a given speed.

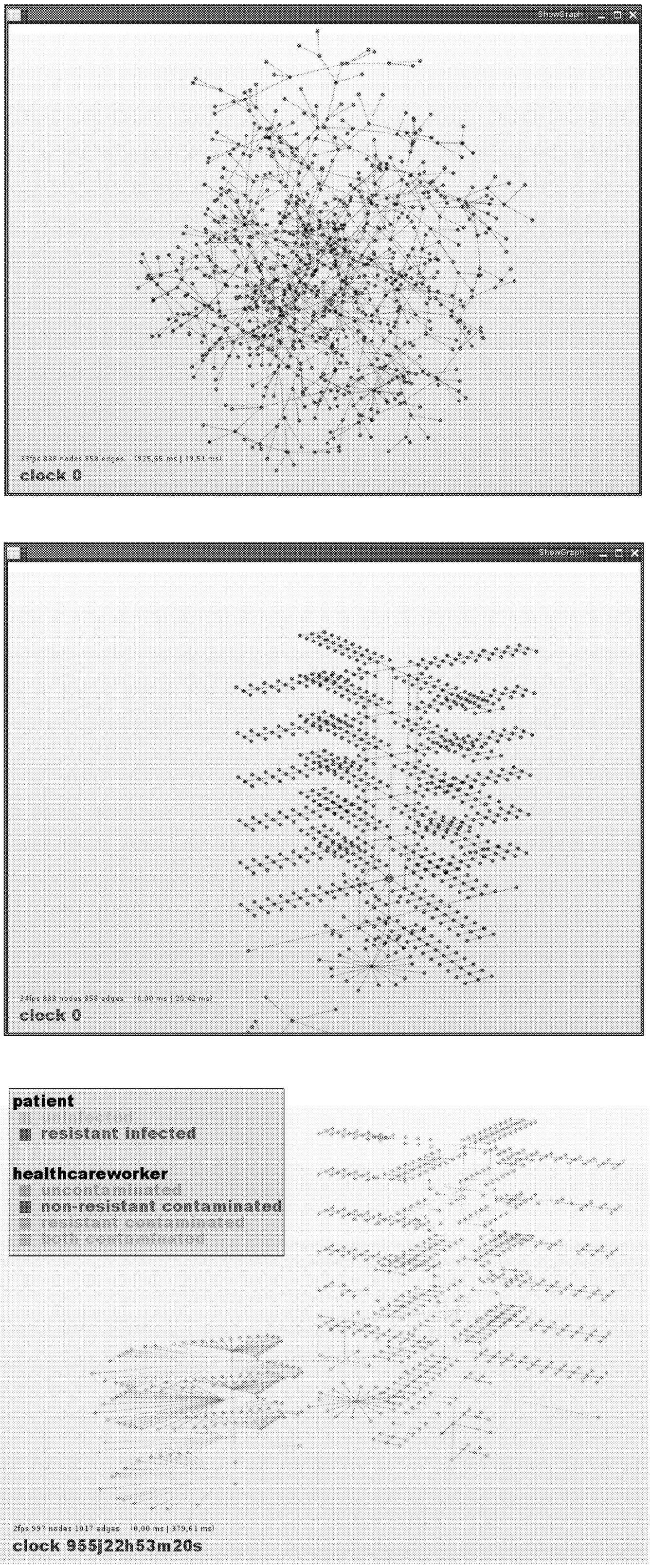

ShowGraph
35fps 838 nodes 858 edges (925,65 ms | 19,51 ms)
clock 0
ShowGraph
34fps 838 nodes 858 edges (0,00 ms | 20,42 ms)
clock 0
patient
infected
▨ resistant infected
healthcareworker
▨ uncontaminated
▨ non-resistant contaminated
▨ resistant contaminated
▨ both contaminated
2fps 997 nodes 1017 edges (0,00 ms | 379,61 ms)
clock 955j22h53m20s

Figure 2: Three Visualizations of the Graphs Representing the Jacques Monod Hospital

SIMULATION

Records of displacement and infection status of each patient, are not easy to maintain. Often only the medical state of the patient is recorded, not its movements. Furthermore as we are interested in nosocomial diseases, it would be interesting to process data concerning healthcare workers, in addition to patients. Indeed, in nosocomial diseases, the hospital plays the role of reservoir of resitant bacterias, and it is the employees that propagate the infection.

In addition, we do not know which sectors of the hospital may eventually become reservoirs for antibiotic resistant bacteria strains. That is, in our model, we often do not know if a vertex is infected or not.

To accommodate the miss of data, the model can be simulated, that is individuals behavior can pre-defined instead of merely reading and interpolating data. Patients and healthcare workers will therefore move inside the hospital going from a starting point to a destination point using routes inside the graph. The way the routes are defined, and how starting and ending points are chosen can be changed.

To define how the infection propagates, an infected individual can infect another by contact. The model allows to specify the contact duration for the contamination to occur. Identically, sectors can be contaminated by individuals if they stay a given amount of time.

As said above, there exist at least two kinds of individuals, patients and healthcare workers. To define how the infection develop, we follow the model developed in [Webb et al., 2005]. In this model, healthcare workers are considered as carrying the infection: they are contaminated by patients and then are able to contaminate other patients. There will therefore be two states for individuals: contaminated and healthy. We also consider two bacteria strains. One is antibiotic resistant, whereas the other does not. When an individual is contaminated he will therefore also contain information about the kind of bacteria he carries.

The bacteria population for each patient evolves according to the following rules that can take into account an antibiotic treatment:

$$\left\{\begin{array}{rcl} \dfrac{dV^-(a)}{da} & = & \left(-\dfrac{\tau V^+(a)}{V^-(a)+V^+(a)}\right. \\ & + & \left.\beta^-(a) - \dfrac{V^-(a)+V^+(a)}{\kappa_F}\right) \\ & \cdot & V^-(a) + \gamma V^+(a) \\ \dfrac{dV^+(a)}{da} & = & \left(\dfrac{\tau V^-(a)}{V^-(a)+V^+(a)}\right. \\ & + & \left.\beta^+(a) - \dfrac{V^-(a)+V^+(a)}{\kappa_F} - \gamma\right) \\ & \cdot & V^+(a) \end{array}\right.$$

Where $V^-(a)$ is the non-resistant bacteria concentration at age a and $V^+(a)$ the resistant bacteria concentration. β^-, β^+, K_F, τ and γ are parameters allowing to regulate the bacteria population evolution.

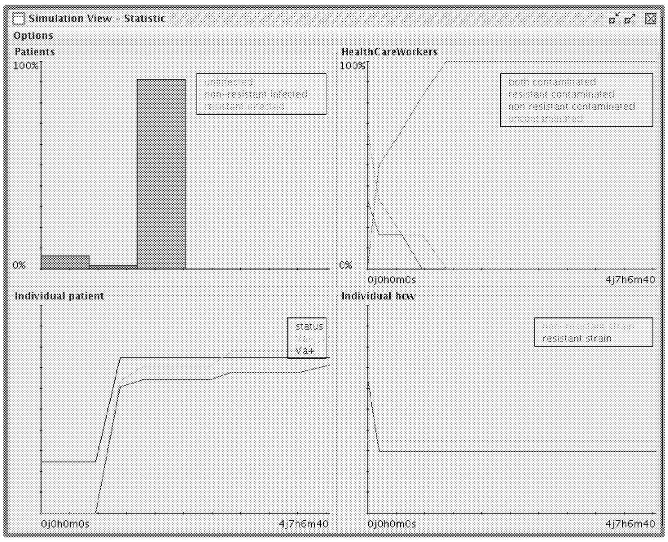


Figure 3: Statistic View of the Simulator

Vertices of the graph can also be of several types. Normal vertices are non contaminated at start and can be infected if contaminated individuals cross or stay on them. Some vertices are considered as sterile and cannot be contaminated. To allow simulations, some vertices can also become contamination reservoirs, that is vertices that are always contaminated. The figure 4 shows such a simulation at distinct time intervals. Each colored vertice represents an infected area.

In addition to the visual representation of the hospital graph, the simulator allows us to follow the development of the infection during the simulation, as shown on figure 3.

USING AN ANT SYSTEM TO DETECT INFECTION AREAS

Given the simulation or replay of an infection using our model, we then search to detect infection paths as well as possible contamination reservoirs. That is, we search sectors where resistants strains of bacteria remain and can contaminate both healthcare workers and patients.

This problem is made more difficult since most of the time the patient infection is not detected as soon as it really occurs. Therefore, not only we have to consider the patient displacements when infected but also before his infection.

We propose to use the ant metaphor to detect the infection paths and reservoirs. The ant metaphor use numerical ants ”travelling in the graph“. The inspiration comes from the observation that real ants are able to detect the better path from their nest to a food source by dropping pheromones on the ground [Dorigo et al., 1996]. Pheromones are olfactory messages that attract other ants. The more of pheromones the more ants are attracted. Indeed, ants having found a good path whatever be the criterion for good, quicker, smarter, easier, will tend to deposit more pheromones on it. This can be implicit, for example shorter path are more quickly full of pheromones, or explicit, for example the ant detects the better path and choose to deposit pheromones on it.

Ants continuously travel in the graph and are continuously able to drop pheromones. An ant travels from vertex to vertex, choosing the next vertex in a balanced random way. That is, the ant chooses the next vertex according to some importance criterion that weight how many chances it has to choose it. In presence of pheromones, ants tend to choose the vertex that have more pheromones, but other criterions may be incorporated in this choice.

Such a behavior leads to more an more ants using important vertices and dropping pheromones on them, without neglecting other vertices, since they only have a larger chance to choose the important ones. Quickly, more and more pheromones are present on important vertices, therefore selecting them. This process increases with time, however pheromones evaporate which tends to avoid over increasing of pheromones values on largely used vertices, and avoid to select unimportant vertex after a long period.

Let's first give a broad idea of our method, inspired by [Dorigo et al., 1996] and [Bertelle et al., 2004]. In our model we can consider two types of "ants". The first one are the individuals inside the hospital. At the contrary of the ant metaphor they do not use information deposited in the graph to choose where to go. However they deposit information in the graph: infected vertices. The second type of ants we propose to add are numerical ants that try to detect infected vertices and paths, that we will call markers.

These later ants will be used at the end of an infection simulation or replay. Their goal is to find paths of infection. For this each ant randomly chooses a entering and destination point, and then explore the graph starting at the entering point, and visiting vertices until it reaches its destination. Each time a marker ant visits a vertex, it memorises it and chooses another vertex. A vertex memory allows to store the path used by the ant, and also to avoid to come back on an already visited vertex. That is, vertex in the ant memory will be avoided unless there is no other possible way (dead-ends for example).

The ant marker algorithm is iterative, at each time step t:

1. all ants move from their source to destination point, this is the "ant-tour";
2. ant paths are evaluated and selected.

The ant-tour process is also iterative. During one time step t of the algorithm, the ant-tour iterate all ants. At each of these ant-tour iterations, ants move from the vertex they are on to another vertex. The ant-tour is iterated until all ants reach their destination. As in the inspiring model, ants will choose the next vertex to visit in a balanced random way, according more chances to be chosen to vertices that are more infected and have more pheromones.

At the end of this displacement process, all ants paths are evaluated. Paths are sorted according to the number and value of infected vertices. Paths whose infection value is lower than threshold ψ are not changed, other paths edges are modified by dropping pheromones on them. This mimic the behavior of natural ants returning to the nest and dropping pheromones to indicate a good path.

This doubly iterative process is repeated until only a limited given number of paths pass the ψ threshold. At this time, the selected paths are considered as infection vectors.

Now we give the details of the method. The set of ants F at each ant-tour is constant. Ants drop pheromones not on vertices bu on edges. This allows to select paths and not only vertices. The pheromone value on an edge (u,v) at time t is:

$$\tau^{(t)}(u,v)$$

Furthermore, at each time step, after the ant-tour, pheromone is added on edges according to ant paths. For an ant x the path is noted W_x and W is the set of all ants paths. The quantity of pheromone dropped by ant x on edge (u,v) at time t is noted:

$$\Delta_x^{(t)}(u,v)$$

and the sum of pheromones dropped by all ants on edge (u,v) at time t is:

$$\Delta^{(t)}(u,v) = \sum_{x\in F} \Delta^{(t)}(u,v)$$

At each time step t, pheromones evaporate. The quantity of pheromones remaining on edge (u,v) is:

$$\tau^{(t)}(u,v) = \rho\tau^{(t-1)}(u,v) + \Delta_x^{(t)}(u,v)$$

With $\rho \in]0,1]$ the pheromone conservation factor.

The probability for an ant on vertex u to cross edge (u,v) choosing vertex v as its next position over all accessible vertices V_u is influenced by the quantity of pheromone on edge (u,v) and the infection value $w^{(t)}(v)$ of vertex v at time t for ant x:

$$\begin{cases} p_x^{(t)}(u,v) &= \dfrac{\left(w^{(t)}(v)\right)^{\beta}}{\displaystyle\sum_{i\in V_u}\left(w^{(t)}(i)\right)^{\beta}} \\ & \text{if } t=0 \\ p_x^{(t)}(u,v) &= \dfrac{\left(\tau^{(t)}(u,v)\right)^{\alpha}\left(w^{(t)}(v)\right)^{\beta}\eta_x(u)}{\displaystyle\sum_{i\in V_u}\left(\tau^{(t)}(u,i)\right)^{\alpha}\left(w^{(t)}(i)\right)^{\beta}\eta_x(i)} \\ & \text{if } t\neq 0 \end{cases}$$

Parameters α and β are used to balance the relative importance of pheromones over the vertex infection. The factor $\eta_x(u)$ allows the ant to minimize the importance of vertices already visited. Indeed, for ant x considering vertex u:

$$\eta_x(u) = \begin{cases} 1 & \text{if} \quad u \in W_x \\ \eta & \text{if} \quad u \notin W_x \end{cases}$$

With $\eta \in]0, 1]$ allowing to minimize the importance of the vertex if it is already in memory $W_x \in W$ of ant x. When the ant-tour finished, the paths W are evaluated. The quantity $\Delta_x^{(t)}$ deposited by ant x on all the edges of its path is:

$$\Delta_x^{(t)} = \frac{W_x}{\max(W)}$$

The paths selected by ants show probable infection paths, but we are also interested at finding possible contamination reservoirs. The method we propose in order to do this, is to superimpose all found paths and detect common vertices. The importance of a vertex as a possible contamination reservoir is then the number of contamination paths crossing it.

CONCLUSION

In this paper a model has been developed that simulates the propagation of infections inside an hospital. The model is individual-based, and allows both the replay of data collected in hospitals, as well as the simulation of an infection propagation. The simulation allows to take into account data that was not or cannot be recorded in real situations. In addition to this model and simulation tool, a method to find infections path and contamination reservoirs inside the hospital that use the ant metaphor has been proposed. The hope of this work is to allow a better understanding of infection propagation and to propose means to stop it and also stop the continuous development of resistant bacteria strains.

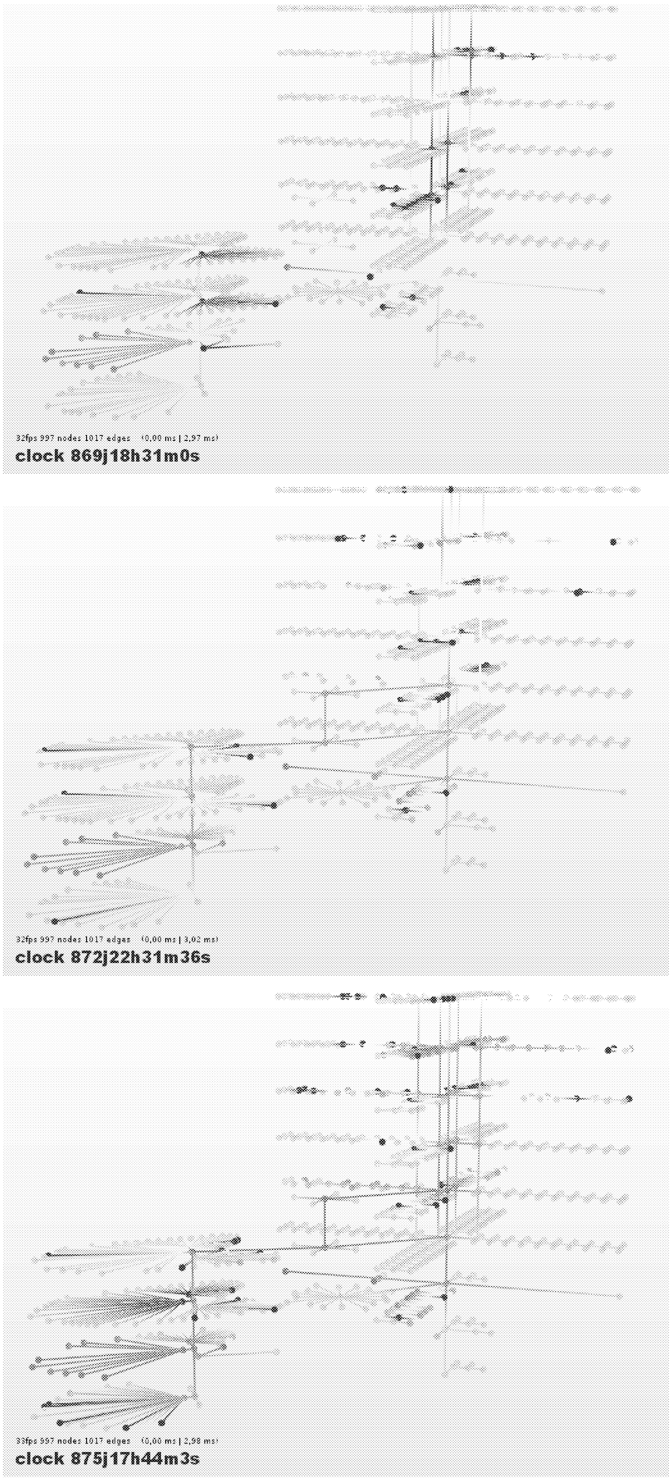
REFERENCES

[Bertelle et al., 2004] Bertelle, C., Dutot, A., Guinand, F., and Olivier, D. (2004). Colored ants for distributed simulations. In LNCS, editor, *ANTS04*, Bruxel. Dorigo et al.

[Dorigo et al., 1996] Dorigo, M., Maniezzo, V., and Colorni, A. (1996). Ant system: Optimization by a colony of cooperating agents.

[Giamarellou, 2002] Giamarellou, H. (2002). Prescribing guideline for severe pseudomonas infections.

[Webb et al., 2005] Webb, G., D'Agata, E., Magal, P., and Ruan, S. (2005). Model of antibiotic-resistant bacterial epidemics in hospitals.



32fps 997 nodes 1017 edges (0,00 ms | 2,97 ms)
clock 869j18h31m0s
32fps 997 nodes 1017 edges (0,00 ms | 3,02 ms)
clock 872j22h31m36s
33fps 997 nodes 1017 edges (0,00 ms | 2,98 ms)
clock 875j17h44m3s

Figure 4: Infection Simulation at Different Time Intervals

ON ADAPTING NEURAL NETWORKS TO CELLULAR MANUFACTURING

Dania A. El-Kebbe, Christoph Danne
Paderborn University / Heinz Nixdorf Institute
Fürstenallee 11, 33102 Paderborn, Germany
Tel.:+49-5251-606495, Fax.:+49-5251-606502
E-mail: elkebbe@uni-paderborn.de

KEYWORDS

Neural networks, Group Technology, Cellular Manufacturing, Part Machine Grouping.

ABSTRACT

This work gives an overview of the neural network based approaches that have been developed to solve the part-machine grouping problem related with the introduction of a cellular manufacturing layout. It also proposes and discusses extensions that should be made to this tool in order to overcome its drawbacks.

INTRODUCTION

In today's business environment, manufacturing companies are constantly searching for improved methods to be able to produce high quality products at low costs. Among the organizational approaches that emerged in past decades, Group Technology (GT) has received considerable attention. In essence, GT seeks to identify similarities among parts and the exploit these similarities in different stages of the production process. The idea is to find groups of parts that to some extend share the same design and manufacturing features and then take advantage of this knowledge in many stages of the manufacturing cycle, like engineering design, process planning, production planning and the production process itself.

The term Cellular Manufacturing (CM) describes a major application of the GT philosophy in the production stage and seeks to bring some of the benefits of mass production to less repetitive job shop manufacturing systems. In contrast to the traditional job shop layout, where similar machines and labor with similar skills are located together, the Cellular Manufacturing approach seeks to form independent cells of machines and labor that are capable of producing one or more groups of similar parts (part families) independently, by that decomposing the manufacturing system into subsystem. The adoption of Cellular Manufacturing has proven to yield some considerable benefits, like reduced setup times, reduced lead times and less material handling.

As the exploitation of similarities among parts forms the basis for GT applications, the detection of these similarities, referred to as *part family formation* and *part classification*, is a prerequisite for any such application. The implementation of a cellular manufacturing system furthermore poses the *part-machine grouping* problem, which describes the task of forming ideally independent machine cells that are able to produce one or more part families autonomously. In practice, it is not always possible to form completely independent machine cells, because the requirement of the same machine by at least two parts from different part families results in inter-cell movements. Such machines are called *bottleneck machines*, the corresponding parts are referred to as *overlapping parts*. A wide range of different approaches has been proposed in past decades to address these problems, including methods based on graph theory [1], matrix sorting [2, 3], mathematical integer programming [4] and expert systems [5]. Most of these approaches have several drawbacks in common, like computational inefficiency when working on industry size datasets and inflexibility when new parts are introduced.

Neural networks offer some unique capabilities that make them extremely suitable for this type of applications. The fact that they can learn by experience to recognize patterns combined with their ability to generalize the knowledge they have obtained are the outstanding advantages that justify the use of neural networks for Cellular Manufacturing applications. Furthermore, when used for clustering, they can classify newly introduced inputs without the need to cluster the entire dataset again. Another advantage is that neural networks are able to handle incomplete data accurately, thus making them useful in real-world applications that have to cope with such information.

APPLICATIONS OF NEURAL NETWORKS TO CELLULAR MANUFACTURING

Neural networks have shown to be a promising tool to address the problems related with autonomic computing, especially such complex tasks with a great number of input elements, because of their greater flexibility and ability to learn from experience and generalize their acquired knowledge to recognize new input patterns. In recent years, much research was dedicated to the application of neural networks for group technology applications, as they are suitable for complex clustering tasks due to their pattern recognition and generalization abilities.

In order to use neural networks to group parts into part families and/or machines into manufacturing cells, it is necessary to find a metric to measure similarity and define how the relevant part features can be coded to serve as input for neural networks. While other Group Technology applications like design retrieval systems [6] analyze the part's design features, almost all approaches for part-machine grouping focus on a part's processing requirements, following an approach called *production flow analysis*

(PFA) [7]. This approach abstains from considering every part feature and focuses on the part's routings, as they imply the part's machine requirements and therefore the relevant information to form independent manufacturing cells. A feasible way to represent machine requirements is via *part incidence matrices*, which are binary-valued matrices consisting of a row for each part and a column for each machine involved. Each entry x_{ij} takes a value of 1 to indicate that part i requires an operation to be performed on machine j, and a value of 0 otherwise. Therefore, each row of the matrix represents the machine requirements of the corresponding part as a binary vector of dimension m, with m being the number of machines considered. These vectors serve as the input for the neural networks.

Part-machine grouping includes the subproblems of forming part families as well as machine cells. Therefore, sequential and simultaneous approaches can be distinguished, dependent on whether they try to solve both problems separately or simultaneously. The neural network based models are sequential approaches, although they can be used to solve both clustering tasks one after another. They can form either part families based on machine requirements (taking the rows of part incidence matrices as their input) or machine cells based on part processing capabilities (taking the columns of part incidence matrices as their input). In the following sections, we assume that part feature vectors form the input and that the network is used to create part families.

Basic network architectures

From the wide range of existing neural network models, some have proven to be especially practical for solving CM related problems. It may be noted that the overwhelming majority of neural networks that have been developed for this task use unsupervised learning methods. This is due to the fact that supervised learning always requires some knowledge about the clusters that will be formed and a set of training data containing input elements with a known correct output value. This type of information is rarely available for part-machine grouping problems, because typically no information about the correct group formation is known a priori.

Nevertheless, Kao and Moon [8] propose a method for part grouping based in a three-layer feedforward network using backpropagation learning (i.e. supervised learning). They try to overcome the lack of training data by arbitrarily selecting some parts (*seed parts*) to be the representatives of the part families and train the network to classify these parts correctly. Subsequently, all remaining parts should be classified by the network to belong to one of these part families. This approach suffers from several shortcomings. First, the task of selecting the seed parts is left to human interaction, thereby depending on a subjective judgment to select parts that are as "distinctive" as possible. This is clearly a weak point, as the selection of representatives for the part families heavily affects the number of part families formed and their characteristics. Second, the training process may be repeated several times with an increasing number of elements in the training set. A newly introduced part that does not fit into any of the existing part families causes the network to create a new part family and start the whole training process again.

Against this background, research has focused on network models using unsupervised learning methods as they make weaker assumptions about the underlying clusters. The basic architecture of all network types considered in the subsequent sections consists of two fully interconnected layers. The input layer contains as many neurons as there are columns in the part incidence matrix (i.e. machines considered) and reads one row of the matrix at a time and passes it to the output layer via weighted connections. Each output layer neuron j represents one cluster, therefore the output is a binary vector with only one entry taking a value of 1 and thus indicating the corresponding part family and values of 0 for all other entries. The output neurons "compete" to respond to the input pattern, computing the weighted sum of their input signals. The winning neuron (i.e. the one with the highest input value) indicates the part family the presented part should be assigned to and updates its weight vector to move closer to the centroid of all elements in this cluster. This is done by transferring a part of the weight of the connections receiving values of 0 at the input layer to the connections that receive values of 1.

This basic mode of operation, called *competitive learning* [9, 10], still suffers from various shortcomings when applied in the context of part-machine grouping. The number of part families created has to be specified in advance and there are no means to control the degree of similarity among the part families. In the following sections, we will outline the more recent approaches that try to overcome these limitations by extending the model in several ways.

Kohonen Self Organizing Feature Maps

Self Organizing Feature Maps (SOFM) [11] may be seen as an extension to the competitive learning model. The special characteristic is a two dimensional output layer, also referred to as the output map or Kohonen layer. The SOFM takes a vector of any dimension as its input and can be used to transform it into a two-dimensional map that can be represented graphically (note that in this case the output neurons do not directly represent part families). Kiang, Kulkarni and Tamb [11] claim that the SOFM performs a dimension reduction of the input patterns to the two-dimensional space while maintaining the topological relations between the elements. Thus, clusters in the higher dimensional input will also appear as clusters in the Kohonen layer. This makes this network particularly useful to be integrated into an interactive decision support system, where the graphical representation of the map can be used by a decision maker to fragment it into the desired number of clusters. In contrast to other neural network based approaches, the SOFM does not perform any grouping of parts itself, it just helps to identify clusters by offering a method to visualize the parts to identify similarities.

The operation of the SOFM is similar to the competitive learning model. The main difference is that not only the weight vector of the winning neuron is updated, but also those of the neurons in a predefined neighbourhood. When an input pattern is presented, for each neuron j in the Kohonen layer, the Euclidean distance of its weight vector to the input pattern:

$$D_j = \sqrt{(x_1 - w_{j1})^2 + (x_2 - w_{j2})^2 + \ldots + (x_m + w_{jm})^2}$$

is computed. Then, the neuron j^* with the smallest distance is selected and its weight vector is updated to move closer to the input pattern according to the equation

$$w_{j^*}^{new} = w_{j^*} + \alpha(X - w_{j^*})$$

where α is the learning rate of the network and controls the speed of weight adoption. To guarantee convergence, it is proposed to start with a high learning rate and a wide neighborhood and decrease both progressively in time.

The Kohonen SOFM have proven to be efficient for part grouping and are a promising approach to be integrated into interactive decision support systems where the final clustering is done by a domain expert.

Adaptive Resonance Theory

Similar to the competitive learning model, the Adaptive Resonance Theory (ART) [12] seeks to classify parts automatically and uses the output layer neurons to directly represent part families. It extends the competitive learning model to overcome two of its biggest drawbacks. First, the parts are not always assigned a part family independently of the degree of similarity. ART neural networks use parameter called vigilance threshold ρ to ensure that similarity within a part family is not less than ρ, based on some similarity measure. Furthermore, the number of part families created does not have to be known a priori, but is determined during the clustering process.

The network architecture is similar to the architecture of competitive learning networks. The number of neurons in the comparison layer thus equals the maximum number of part families expected. Associated with each output neuron j, there is a weight vector $W_j = (w_{j1}, w_{j2}, \ldots, w_{jm})$ and an exemplar vector $T_j = (t_{j1}\ t_{j2}, \ldots, t_{jm})$. The exemplar vector is a binary representation of the weight vector, thereby representing the characteristics of the corresponding part family. Furthermore, recurrent connections are introduced, so that an input layer neuron I is connected to an output neuron j via a feedforward and a feedback connection with connection weights w_{ji} and t_{ji}, respectively.

For a better readability of the algorithm described below, we define $\|X\| = \sum x_i$. Note that for a binary vector X, this is simply the number of '1's in vector. Furthermore, let the intersection of to vectors $X \cap Y$ denote the vector C, whose elements are obtained by applying the logical AND operator to the corresponding elements in X and Y, implying that $c_i = 1$ if $x_i = y_i = 1$, and 0 otherwise.

1. Initialization: Select vigilance parameter ρ in the range [0,1]. Initialize weight vectors with entries $w_{ji} = 1/(1 + m)$ and exemplar vectors with entries $t_{ji} = 1$ for all i, j.
2. Present an input pattern $X = (x_1, x_2, \ldots, x_m)$. For each output neuron j, compute $net_j = W_j \bullet X$ (weighted sum of inputs).
3. Select output node j* with the highest netj value. The exemplar vector Tj* of this neuron is fed back to the input layer. To ensure similarity is higher than the threshold, check if
$$\frac{\|X \cap T_{j^*}\|}{\|X\|} > \rho$$
4. If similarity check fails, disable node j^* temporarily for further competition (to avoid persistent selection) and return to step 3.
5. If the similarity check is successful, set output value for j^* to 1 and 0 for all other output neurons. Update exemplar vector T_{j^*} to $\|X \cap T_{j^*}\|$. Furthermore, update weight vector W_{j^*} according to the equation
$$w_{j^*i} = \frac{x_i \wedge t_i}{0{,}5 + \|X \wedge T_{j^*}\|}$$
6. Enable any nodes disabled in step 5. Repeat steps 2.-7. until the last input pattern has been presented.
7. Repeat the entire process until the network becomes stable, meaning that the weight vectors stabilize to fixed values with only small fluctuations.

According to this algorithm, the ART network determines the cluster that is most similar to the current input pattern. Then, the similarity between the input and the cluster is computed as the fraction of perfectly matching '1's in the input and the exemplar vector in relation to the total number of '1's in the input vector. If this similarity measure is above the vigilance threshold, the input is assigned the corresponding class and the cluster's weight and exemplar vectors are updated to represent the new element. If the similarity check fails, the procedure is repeated with the output neuron with the next highest *net* value. If the input can not be assigned any class, a yet uncommitted output neuron is selected eventually to represent this new class and its weight- and exemplar vectors are updated accordingly.

In the steady state, the connection weights also provide valuable information for the subsequent machine cell formation. A strictly positive connection weight w_{ji} implies that part family j contains at least one part that requires an operation to be performed on machine i. Therefore, considering the connection weight matrix can be used to easily assign machines to part families and also detect bottleneck machines and overlapping parts.

The ART network model provides the possibility to control the similarity among parts within one part family via the vigilance threshold and also does not require the number of part families to be specified in advance. Nevertheless, it also suffers some drawbacks. The biggest problem related to the traditional ART model is the *category proliferation problem* [13]. During the clustering process, a contraction of the exemplar vectors can be observed, due to repeatedly forming the intersection with newly assigned input patterns. This leads to sparse exemplar vectors, which causes the network to create new categories frequently, as the number of unsuccessful similarity checks increases. Kaparthi and Suresh [13] found out that because of this problem clustering is more precise when the density of the part-machine incidence matrices is high. As density is usually low, they propose to inverse the matrices for clustering purpose. This approach was further investigated and the improvements it brings to performance were confirmed by Kaparthi et. Al. [14]. Dagli and Huggahalli [12] propose not to store the intersection of input- and exemplar vector, but the one that has the higher number of '1's. Chen and Cheng [15] state that the above methods may lead to improper clustering results and develop advanced preprocessing techniques to prevent category proliferation.

Another shortcoming is that clustering with ART networks is sensitive to the order in which the inputs are presented. Dagli and Huggahalli [12] propose to preprocess the input vectors and present them in order of descending number of '1's, which also would help to address category proliferation, as the exemplar vectors would initially be taken from the inputs with dense feature vectors. Apart from the order of the inputs, clustering is very sensitive to the choice of the vigilance parameter. While it can be seen as an advantage to have the ability to control the size and number of clusters formed, it is also difficult to find the value that yields the best results.

Fuzzy Adaptive Resonance Theory

The Fuzzy ART model is the most recent development of neural networks for part-machine grouping. It tries to improve ART neural networks by incorporating the concepts of fuzzy logic. This model introduced by Carpenter, Grossberg and Rosen [16] and has also been applied to the cell formation problem in several works. For example, Suresh and Kaparthi [17] use a Fuzzy ART network to form part families based on part-incidence matrices and compare their performance to matrix manipulation algorithms like ROC [18] and also traditional ART, showing that Fuzzy ART networks are superior to these methods.

A fundamental difference of fuzzy ART networks in comparison with ART is the fact that they can handle both binary and non-binary input data. In order to incorporate fuzzy concepts into ART networks, occurrences of the intersection of two vectors applying a logical AND on the corresponding elements are replaced by the fuzzy AND operator $\wedge$, defined as $(x \wedge y) = \min(x,y)$ [16]. In essence, the algorithm operates just as the ART network, with the following differences:

- The input of each output neuron j is computed as
$$net_j = \frac{\|X \wedge W_j\|}{\alpha + \|W_j\|}$$
- When checking for similarity, the fuzzy AND operator is used:
$$\frac{\|X \wedge W_{j^*}\|}{\|X\|} > \rho$$
- The weight update changes to incorporates the learning rate:
$$W_{j^*}^{new} = \beta(X \wedge W_{j^*}^{old}) + (1-\beta)W_{j^*}^{old}$$

With the use of the fuzzy operator, the exemplars of the clusters are not restricted to binary values, and there is no need for a binary exemplar vector. Instead, both the bottom-up and the top-down connection have a weight denoted with w_{ji}, thus the weight- and exemplar vector of each output neuron are identical. In addition to the vigilance threshold, two more parameters are introduced, viz. the choice parameter α and the learning rate β. The learning rate specifies the speed with which the exemplars are updated in response to new inputs, which can be used to control the speed of learning and thereby reduce category proliferation.

Suresh and Kaparthi [17] show that for the cell formation problem, applications using Fuzzy ART networks perform better and yield more consistent clustering results. This is especially due to the generalized update rule that can be controlled via the learning rate parameter. During the update process in ART networks, an entry was set to zero if the newly assigned pattern did not have this feature, which lead to rapid contraction of the exemplar vectors and thereby to category proliferation. In Fuzzy ART, the learning rate controls to what extend the exemplar entries are adjusted. For example, given a learning rate of $\alpha = 0.1$, an exemplar entry $w_{ji} = 1$ is adjusted to 0.9 the first time an input pattern without the corresponding feature is assigned to that cluster. In ART models, it would have been set to zero, reducing the probability of a pattern with that feature being assigned to that class significantly, and causing category proliferation in the long term. Note that a Fuzzy ART model with a learning rate of one and binary input vectors operates the same way as an traditional ART network.

The learning rate can also be adjusted to take advantage of another optional feature of Fuzzy ART networks, called *fast-commit slow-recode* [16]. This option involves a dynamic change to the learning rate that distinguishes between the first assignment of a pattern to a yet uncommitted node and the assignment to an existing cluster. In the first occurrence of a pattern, the learning rate is set to one to make the exemplar of the new cluster match the input vector. For subsequent assignments to that cluster, the updating process is dampened by choosing $\alpha < 1$. This method can help to incorporate rare features quickly in a new vector and prevent the quick deletion of learned pattern because of partial or noisy input data.

The Fuzzy ART model can be used for part family and cell formation in the same way as the ART model. But it provides several advantages over traditional ART, like the ability to handle continuous inputs, although the simple cell formation methods use binary part-incidence matrices as their input and thus do not yet take advantage of this ability. Furthermore, Fuzzy ART models provide more means to address the category proliferation problem, which is also critical in the context of group technology, as it is desirable to control the number of part families or machine cells created. The improved clustering precision is also due to the adjustable weight updates that prevent fast contraction of exemplar vectors and thereby increase the probability that similar input patterns will be grouped together, even when the corresponding vector has already been updated in between.

Because of their promising results, Fuzzy ART networks are subject of current research effort. The work by Suresh and Park [22] introduces an extension to Fuzzy ART networks that allows them to consider part operation sequences during the clustering process. Peker and Kara [23] recently investigated on parameter setting for Fuzzy ART networks.

DIRECTIONS FOR FUTURE WORK

Although neural network algorithms have been improved constantly to solve the clustering problem more efficiently, little effort has been made to solve the problem of bottleneck machines and overlapping. The majority of the methods implicitly duplicates machines and accepts overlapping parts that appear. Neural networks alone can not decide if it is advisable to duplicate a machine, because there is a more complex cost benefit analysis involved that requires information not available in the network. But due to the fact that many of the methods presented can be used to detect bottleneck machines, they may be integrated into decision support systems to help a decision maker or even be combined with other intelligent methods (e.g. expert systems) that are able to perform a cost benefit analysis.

Considerable research effort has been made to solve the unconstrained part-machine grouping problem. Currently, Fuzzy ART appears to be the most promising approach. Nevertheless, it currently uses fuzzy concepts only within the network and not yet at the input or output level, which could be a promising area for further research [19]. For instance, the network could indicate the degree of membership to each family with continuous values.

Although the neural network models have become more sophisticated over the years, there are still several challenges that have not received sufficient attention in literature so far. The methods described in this work focused on the unconstrained part and / or machine grouping based on the part's processing information or part features. In real world applications, several additional constraints have to be considered when designing a cellular factory layout. For instance, it may be necessary to consider part demands, capacity constraints or load balancing. Thanks to the development of more sophisticated clustering methods like Fuzzy ART networks and improved computational possibilities, these problems can be addressed in current and future research.

Rao and Gu [20] point out that the assumptions of unlimited capacity can lead to nonimplementable solutions and suggest an extended unsupervised network similar to the ART model to handle machine availability and capacities as additional constraints. This is mainly done by introducing a hidden layer with nodes representing each machine type that contain knowledge about the number of machines available and their capacity in the form of threshold functions. Parts can only be assigned to a part family if it does not violate any of these constraints.

Clearly neural networks will not be able to handle all constraints alone, but must be integrated with other methods. Some research effort has already been made in this area. For example, the work of Suresh and Slomp [21] proposes a framework to include these additional constraints into the cell formation process. Theiy use a Fuzzy ART network first to form part families, before a mathematical goal programming model taking into account the additional constraints is used to form the corresponding manufacturing cells. Another interesting aspect of this work is the proposal to integrate this framework into a decision support system. As the execution time of the clustering process is fairly low by now, it is possible to have human interaction. This could help to make the complex decisions that require knowledge not available in the system, like considering the acquisition of new machines opposed to accepting inter-cell movements.

Therefore, it is likely that future research will focus on extending the use of fuzzy concepts, integration of supplemental methods to consider more complex constraints and integration into decision support systems.

CONCLUSIONS

In this work, we gave an overview of neural network based methods proposed to address the problems related with the introduction of a cellular manufacturing layout. Neural networks have shown to be a promising tool for such complex clustering tasks with a great number of input elements, because of their greater flexibility and ability to learn from experience and generalize their acquired knowledge to recognize new input patterns. Current Fuzzy ART network provide consistent clustering results and an extension of the use of fuzzy concepts, as well as the integration of neural networks with other approaches appear to be promising research areas for the future.

REFERENCES

[1] R. Rajagopalan & J.L. Batra**,** Design of cellular production systems: A graph theoretical approach, *International Journal of Production Research*, *13*(6), 1975, 567-573.

[2] J.R. King, Machine-component grouping in production flow analysis: an approach using a rank order clustering algorithm, *International Journal of Production Research*, *18*(2), 1980, 213-232.

[3] J.R. King & V. Nakorncha, Machine-component group formation in group technology-review and extension, *International Journal of Production Research*, *20*(2), 1982, 117-133.

[4] A. Kusiak, The generalized group technology concept, *International Journal of Production Research*, *25*(4), 1987, 561-569.

[5] A. Kusiak, EXGT-S: A knowledge-based system for group technology, *International Journal of Production Research*, *26*(5), 1988, 887-904.

[6] V. Venugopal & T.T. Narendran, Neural network model for design retrieval in manufacturing systems, *Computers in Industry*, *20*, 1992, 11-23, 1992.

[7] J.L. Burbridge, Production flow analysis for planning group technology, *Journal of Operations Management*, *10*(1), 1991, 5-27.

[8] Y.B. Kao & A.Moon, A unified group technology implementation using the backpropagation learning rule of neural networks, *Computers & Industrial Engineering*, *20*(4), 1991, 435-437.

[9] V. Venugopal & T.T. Narendran, Machine-cell formation through neural network models, *International Journal of Production Research*, *32*(9), 1994, 2105-2116.

[10] C.H. Chu, Manufacturing cell formation by competetive learning, *International Journal of Production Research*, *31*(4), 1993, 829-843.

[11] M.Y. Kiang, U.R. Kulkarni & K.Y. Tamb, Self-organizing map network as an interactive clustering tool - An application to group technology, *Decision Support Systems*, *15*(4), 1995, 351-374.

[12] C. Dagli & R. Huggahalli, Machine-part family formation with the adaptive resonance theory paradigm, *International Journal of Production Research*, *33*(4), 1995, 893-913.

[13] S. Kaparthi & N.C. Suresh, Machine-component cell formation in group technology-a neural network approach,

International Journal of Production Research, *30*(6), 1992, 1353-1368.

[14] S. Kaparthi, N.C. Suresh & R.P. Cerveny, An improved neural network leader algorithm for part-machine grouping in group technology, *European Journal of Operational Research*, *69*(3), 1993, 342-356.

[15] S.-J. Chen & C.-S. Cheng, A neural network-based cell formation algorithm in cellular manufacturing, *International Journal of Production Research*, *33*(2), 1995, 293-318.

[16] G.A. Carpenter, S. Grossberg & D.B. Rosen, Fuzzy ART: Fast stable learning and categorization of analog patterns by an adaptive resonance system, *Neural networks*, *4*, 1991, 759-771.

[17] N.C. Suresh S. & Kaparthi, Performance of Fuzzy ART neural network for group technology cell formation, *International Journal of Production Research*, *32*(7), 1994, 1693-1713.

[18] J.R. King, Machine-component grouping in production flow analysis: an approach using a rank order clustering algorithm, *International Journal of Production Research*, *18*(2), 1980, 213-232.

[19] V. Venugopal, Artificial neural networks and fuzzy models: New tools for part-machine grouping, In: N.C. Suresh & J.M. Kay (Eds.): *Group Technology and Cellular Manufacturing: State-of-the-art Synthesis of Research and Practice* (Kluwer Academic Publishers, Norvell, Mass, 1998).

[20] H.A. Rao & P. Gu, A multi constraint neural network for the pragmatic design of cellular manufacturing systems, *International Journal of Production Research*, *33*(4), 1995, 1049-1070.

[21] N.C Suresh & J. Slomp, A multi-objective procedure for labour assignments and grouping in capacitated cell formation problems, *International Journal of Production Research*, *39*(18), 2001, 4103-4131.

[22] N.C. Suresh & S. Park, Performance of Fuzzy ART neural network and hierarchical clustering for part–machine grouping based on operation sequences, *International Journal of Production Research*, *41*(14), 2003, 3185-3216.

[23] A. Peker & Y. Kara, Parameter setting of the Fuzzy ART neural network to part–machine cell formation problem, *International Journal of Production Research*, *42*(6), 2004, 1257–1278.

EMOTION MODELLING

SIMULATION OF EMOTIONAL PROCESSES IN DECISION MAKING

Karim Mahboub
University of Le Havre, LITIS Laboratory
25, rue Philippe Lebon, B.P. 540
76058 Le Havre Cedex France
Karim.Mahboub@gmail.com

KEYWORDS

Emotion, decision making, gambling task, behavioral graph.

ABSTRACT

Human emotional capabilities have recently been considered essential in decision making. An emotional model applied to the *Gambling Task* game is outlined here. The aim is to be able to simulate a human behavior with respect to the emotional feedback created by the environment. To do so, an OCC (Ortony, Clore and Collins) model emotion type is used to define several criteria representing a human emotional structure. Moreover, a probabilistic graph is brought into play for the behavior representation through the game environment. Results are promising and show that the emotional reactivity is coherent and globally lead the player to his objectives.

INTRODUCTION

This study was brought to life from the assumption that emotion could be part of human intelligence and particularly be of a great help in decision making. Indeed, Damasio's works [Bechara et al., 2000] have proved that with a lack of emotional activity, a human being is hardly able to make a reasonable decision when facing a problem.

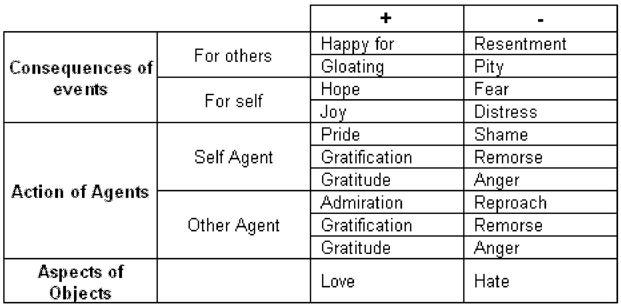
The objective of this work is to simulate emotions using a sociological approach within a very particular context in the world of cognitive psychology: the *Gambling Task.* To do so, we worked together with the *Psychology and cognitive neurosciences laboratory* of Rouen (France). This collaboration will allow us to draw a parallel between human behaviors and computed simulation outcomes.

After having had a look at the sociological aspects of emotions, we will develop a model in response to the problem of emotions simulation. Finally, the results obtained and the planned perspectives will be presented.

STATE OF THE ART

When looking at the different definitions of emotion on the internet or in any kind of book, we realise how difficult it is to comprehend all the aspects involved in emotional processes. Hence, several sociologists and psychologists have tried to make a list of all the possible human emotions. Most of them have finally determined a few basic emotions that are considered being sufficient to represent any human feeling. For instance, Ekman, a well-known American psychologist, describes 6 basic emotions: anger, disgust, fear, joy, sadness, surprise.

Apart from the actual distinction of these emotional concepts, psychologists, from their point of view, tried to make different types of categorisations for emotions. One of the most famous is the *OCC model*, made by three psycho-cogniticians: Ortony, Clore and Collins [Ortony et al., 1988].



		+	-
Consequences of events	For others	Happy for	Resentment
		Gloating	Pity
	For self	Hope	Fear
		Joy	Distress
Action of Agents	Self Agent	Pride	Shame
		Gratification	Remorse
		Gratitude	Anger
	Other Agent	Admiration	Reproach
		Gratification	Remorse
		Gratitude	Anger
Aspects of Objects		Love	Hate

Figure 1: *The OCC Model*

According to these three authors, emotions come from the appraisal of three different aspects of the world: the consequences of events, the actions of agents and the perception of an object. For instance, an event which aims at realising a particular objective will create joy; an action from an agent (individual) that would go against his principles will end up with a feeling of shame; the perception of an object can be disgusting depending on the agent preferences.

Therefore, the OCC model (see figure 1) defines three classes of emotions, according to the emotional context they refer to, in which we have smaller groups concerning the person responsible for the triggering of the action (generally 'other' and 'oneself'). Each emotional dimension is represented by an antagonistic couple, like *joy* and *distress*, or *love* and *hate*, in which an individual's emotional state is located, somewhere between the two bounds.

MODEL OF EMOTION

Within the framework of this project, the modelisation which is used is based on the OCC system described above. For each antagonistic couple previously listed, we associate a fuzzy logic sigmoid function called μ, defined on the interval $[-1;1]$, as follows:

$$\begin{array}{cccc} \mu: & [-1;1] & \mapsto & [0;1] \\ & k & \rightarrow & \mu(k) \end{array}$$

The variable named k represents the current position of the emotional criterion on the curve. According to its value, the actual emotional criterion is, more or less, positive or negative. Therefore, if k equals 0, the emotional state related to the couple *joy-distress* is neutral. The more k is close to -1, the more important the feeling of distress, and vice versa, the more k is close to 1, the more important the feeling of joy.

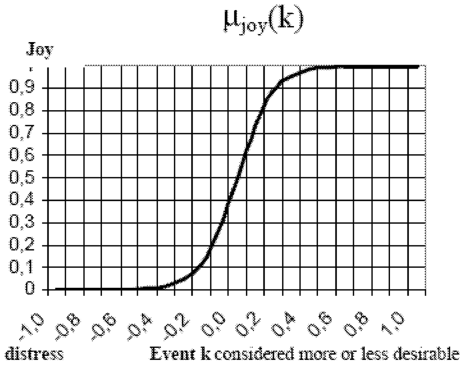


Figure 2: *The μ_{joy} function*

The shape of this sigmoid function can vary, with respect to the feeling couple which is modelised [Colloc and Bertelle, 2004]. However, it can also change depending on the actual individual involved. As a result, an easily depressed person will have a very steep *joy-distress* curve, with the ability to go quickly from a great joy to a deep sorrow. On the other hand, a mentally strong person will end up with a gentle curve.

Here is the sigmoid function used in the model:

$$\mu(k) = \frac{1}{(1 + e^{-\lambda k})} \qquad (1)$$

The λ value is proportional to the curve slope.

Gambling Task

The Gambling Task (see Bechara and Damasio's works [Bechara et al., 2000]) is a well-known test which aims at stimulating the emotional processes of a player by giving him some money and proposing him to increase his capital through a card game.

This game consists of four decks of cards. On each card a number representing a certain amount of money, positive or negative, is written. For each card taken, the player wins or looses money with respect to the amount indicated on the card. The game stops when a hundred cards have been taken.

Of course, in order to make the game more interesting, the player is told that amongst the four packets, two of them are more profitable than the other two. Indeed, the average sum of all the cards located in a profitable packet is positive, meaning that the player has more chance to earn money. However, the sums written on these cards are not high. Eventually, the profitable packets make the player earn small amounts of money without loosing much. On the other hand, the bad packets are not profitable at all, and make the player loose a lot, even more than he will ever earn on these packets.

Finally, the player's goal is to collect money by identifying the good packets. He will then unconsciously develop a strategy to achieve this goal.

Probabilistic behavioral graph

During the game, the player obeys a behavioral graph that describes the environmental protocol. In other words, each packet is represented by a state in the graph, and a connection between two states is a choice that the player made to go from a packet to another.

In the following example (figure 3), the player has just taken a card from deck 1 and is planning to choose the next card. He has four possible choices: staying in packet 1 or changing for packet 2, 3 or 4.

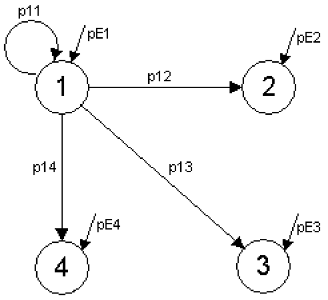


Figure 3: *The behavioral graph*

Each possible choice is linked with a particular probability p, calculated from the previous results obtained in the four decks. Hence, the sum of the four probabilities stemming from the first deck must be 1.

The p_{En} are the entry values for each state (used only to choose the first card of the game).

Decision making process

Decision making consists in :

1. Defining the values in the behavioral graph according to the player emotional state and the history of every deck of cards. To do so, we need to calculate every probability value P in the graph, knowing the emotional $\mu_i(k_i)$ curves (i being an emotional couple), and using valence function α_i as follows :

$$\forall transition\ t\ \forall criterion\ i \left\{ \begin{array}{ll} p_i = & \alpha_i(t)\mu_i(k_i) \\ P_t = & \sum p_i \end{array} \right. \qquad (2)$$

2. Calculating the score of every deck, using the gains and losses that occurred so far (the history of each deck). The

default equation used is :

$$Score = \frac{(gains \times gainMax) - (losses \times lossMax)}{(gains \times gainMax) + (losses \times lossMax)} \tag{3}$$

The $gains$ and $losses$ values correspond to the sum of all the positive and negative cards that have been taken on the deck, $gainMax$ and $lossMax$ being the best and the worst card taken. When calculated, the $score_i$ value is then multiplied by the probability value P_i, in order to influence the latter positively or negatively.

3. Normalising the P_i values, so that their sum equals 1.

4. Choosing randomly a packet in which the player is going to take a card, according to the P_i values.

Emotional feedback

Emotional feedback is the effect of decision over the mind. In the Gambling Task, this phenomenon depends on the game situation, described by the last card taken. The influence of the card over the player's emotion is composed of different parameters.

Firstly, we assume that the intensity of the emotion is directly proportional to the value of the card taken, positively or negatively.

Secondly, we need to take into account the capital of the player at the time he takes the card. Indeed, we can easily consider that a card indicating a strong loss will not especially have a big influence if the player has a lot of money at that time.

Finally, we consider that the emotional state of a normal human being can not be strongly modified, in a realistic manner. Therefore, the emotional feedback will not be computed directly through an equation but using a differential modification of the concerned parameters, with the idea that the emotional state is to be changed gradually, without abrupt transformation.

Considering all these assumptions, we obtain the following equation:

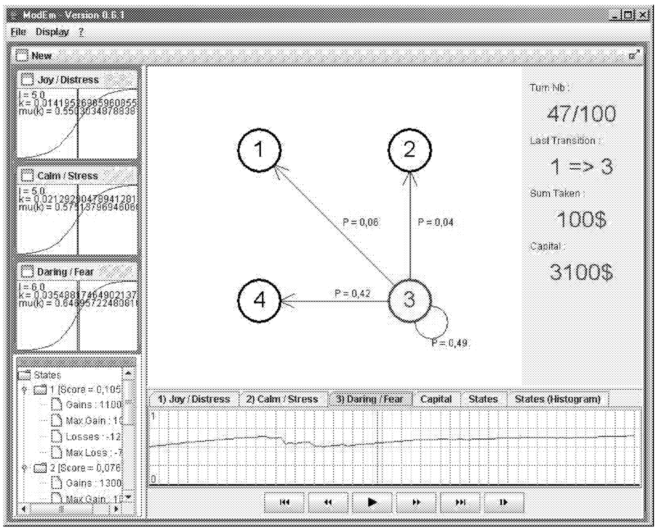
$$k_i = k_i + \frac{v}{c} \times r \times \alpha_i \tag{4}$$

i : index of the emotional criterion to be modified;
k_i : value of the emotional criterion curve on the X axis;
v : value of the card taken;
c : player capital just before the draw;
r : "mind resistance" coefficient ($0 \leq r \leq 1$);
α_i : magnitude of the emotional criterion compared with the other criteria.

The r coefficient represents the player emotionalism, defined in the interval $[0; 1]$. It is his ability to keep his calm in high-rate emotional situations. The more this value is closed to 1 and the more sensitive the player is, and vice versa.

IMPLEMENTATION AND RESULTS

ModEm is an application that directly implements the emotional model previously seen. It draws a probabilistic graph representing the Gambling Task environment and shows all the results related to the emotional state of the player as well as his situation in terms of money and decision parameters.



Turn Nb : 47/100
Last Transition : 1 => 3
Sum Taken : 100$
Capital : 3100$

Figure 4: *The graphical user interface*

Moreover, it uses a special file format which describes the entire game protocol:

- The global parameters: number of turns and starting capital;

- The composition of the packets: number of packets, size and description of all the cards;

- The emotional system: emotional model used, emotional resistance, λ values for each curve, and starting k values.

```
*************************************
#ModEm - Version 0.6.1
*************************************

#TURN_COUNT 100
#CAPITAL 2000
#PACKET_SIZE 40
#PACKET_COUNT 4
#PACKET_CARDS 100 100 -500 ...
#PACKET_CARDS 100 100 -1250 ...
#PACKET_CARDS 50 50 -150 ...
#PACKET_CARDS 50 50 -250 ...
#FORCE_STATE 0 (1)
[#FORCED_STATES 0 2 4 3 1 ...]

#EMOTIONAL_MODEL KM
#RESISTANCE 0.5
#LAMBDA_VALUES 5.0 5.0 6.0
#K_VALUES 0.0 0.0 0.0

#ALPHA_VALUES 0.2 0.3 0.5
#PROBA_VALUES 1.0 1.0 1.0
#PROBA_CHOICE 1
#SCORE_CHOICE 1
```

Finally, the application is able to store its information, with the aim to allow the user to reach any previous played turn in the same game. It can also force the player to choose a particular packet for each turn by using a specified sequence of packet numbers written in the file.

Results

In this part, we will have a look at the obtained curves and graphs in order to analyse the global coherence of the application.

The behavioral graph initially describes four states corresponding to the four decks of cards, each state being valued 0.25, i.e. 25%. This is coherent since the player, before taking his first card, has no information about the decks. Therefore, their probabilities are equal.

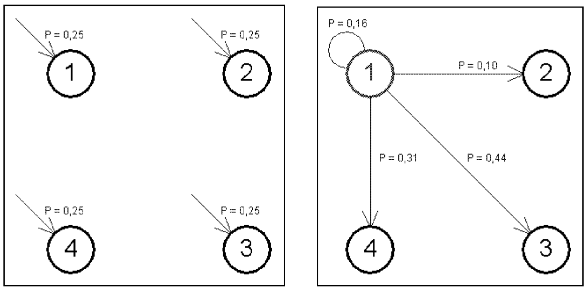


Figure 5: *Example of game graph (start and end)*

At the end of the simulation, probabilities change, allowing the game to distinguish the beneficial packets from the unfavorable ones. Hence, in figure 5, we clearly see that decks 1 and 2 (the disadvantageous packets) have respectively 16% and 10% of chance to be selected, wether decks 3 and 4 (the advantageous packets) have 44% and 31%.

The application also produces behavioral curves that give information about the evolution of the game and the emotional state of the player.

Emotional curves (see figure 6) are defined between 0 and 1 and show the different values of the $\mu(k)$ function according to time.

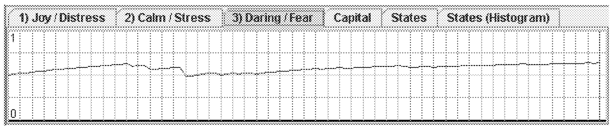


Figure 6: *Emotional curve example ($\mu(k)$)*

The capital curve (see figure 7) helps to see the evolution of the capital throughout the game.

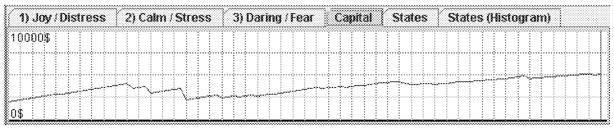


Figure 7: *Capital curve example*

Finally, the state curve (see figure 8) gives details about the different choices of packets. It can either be seen as a curve, or in the form of a histogram.

We can notice that the player has taken much more cards in packets 3 and 4 (the good ones), which means that he is naturally more attracted by these packets. This behavior is rather close to reality since a standard human being who plays the Gambling Task usually reacts this way.

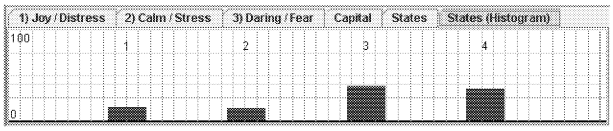


Figure 8: *State histogram example*

CONCLUSION AND PROSPECTS

In order to improve the player environmental comprehension, we aim at completing the model structure using cognitive agents. For a better adaptation to more important problems, we need to add a cognitive module that will be responsible for the player reasoning abilities and also memory issues.

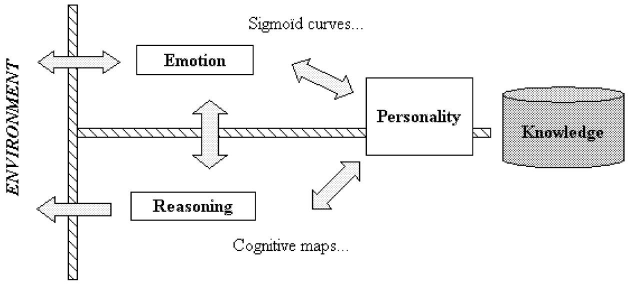


Figure 9: *Mental representation of an emotional cognitive agent*

In order to simulate cognitive capabilities, we use Fuzzy Cognitive Maps (FCM, see [Axelrod, 1976]). These maps are an adaptative manner of learning new knowledge from the environment by adding new states in the behavioral graph. Thus, this cognitive approach allows the player to develop his own personality through his experience stored in his knowledge base representing his memory. The emotion is then considered as a base layer for the decision making procedure, synchronised with the cognitive module. This evolutionary model can not only allow a growth of intelligence through experience, but also communication between agents, with the aim to share knowledge, feelings or points of view.

A second aspect which allows a better understanding is the validation of the application with the help of the experiments made in the *PSY.CO* laboratory in Rouen by comparing human nervous intensity measures with emotional behavior curves obtained within the program.

The OCC model, one of the most famous emotional representation created in the domain of sociology, has been successfully implemented in a computer simulation. The example of decision making through the Gambling Task context shows that emotion is a wonderful driving force which strongly leads the player with his choices. Results are consistent and globally correspond to a real human behavior when making a decision involving stress through the gain or loss of money.

However, emotion is a very complex entity, with many different parameters to be taken into account. So far, we used the OCC model to simulate emotion, but this model only consists of a list of basic emotions within a sociological point of view. There is no deep understanding of the origin and the evolution of emotional processes, and sociology usually has a meta-comprehension of human cognitive activity. In order to create an accurate simulation of human emotion, we need to take care of neurological issues and study the basics of cerebral organisation. This is probably the future of emotion simulation.

REFERENCES

[Hum,] The HUMAINE Website (Human-Machine Interaction Network on Emotion) `http://emotion-research.net/`.

[André, 2002] André, E. (2002). State of art of emotion and personality. Presentation at Virtual Humans Workshop. `http://www.ict.usc.edu/~vhumans/presentations/Andre-Emotions.ppt`.

[Axelrod, 1976] Axelrod, R. (1976). *Structure of decision*. Princeton University Press, Princeton, New Jersey.

[Bechara et al., 2000] Bechara, A., Damasio, H., and Damasio, A. (2000). Emotion, decision making and the orbitofrontal cortex. *Cerebral Cortex*, Volume 10:295–307. `http://cercor.oxfordjournals.org/cgi/content/full/10/3/295`.

[Becheiraz and Thalmann, 1998] Becheiraz, P. and Thalmann, D. (1998). A behavioral animation system for autonomous actors personified by emotions. `http://citeseer.ist.psu.edu/becheiraz98behavioral.html`.

[Chittaro and Serra, 2004] Chittaro, L. and Serra, M. (2004). Behavioral programming of autonomous characters based on probabilistic automata and personality. *Journal of Computer Animation and Virtual Worlds*, Volume 15(Issue 3-4):p 319–326.

[Colloc and Bertelle, 2004] Colloc, J. and Bertelle, C. (2004). Multilayer agent-based model for decision support system using psychological structure and emotional states. *Proceedings of ESM'2004*, pages p 325–330.

[Damasio, 1995] Damasio, A. (1995). *Descartes' Error: Emotion, reason and the human brain*. Harper Perennial.

[Damasio, 2003] Damasio, A. (2003). *Spinoza avait raison - Joie et tristesse, le cerveau des émotions*. Editions Odile Jacob.

[Derre, 2004] Derre, M. (2004). Modélisation à l'aide d'une approche multi-agents de l'émotion, de la structure psychique, de la pathologie dans la prise de décision. `http://www-lih.univ-lehavre.fr/~bertelle/stagesDeaIta2004/rapport_derre.pdf`.

[El-Nasr et al., 2000] El-Nasr, M., Yen, J., and Loerger, T. (2000). Flame fuzzy logic adaptive model of emotions. *Autonomous agents and Multi-agent systems*, Volume 3:p 219–257. `http://citeseer.ist.psu.edu/394133.html`.

[Ferber, 1995] Ferber, J. (1995). *Les systèmes multi-agents - Vers une intelligence collective*. InterEditions, Paris.

[Howard and Howard, 1995] Howard, P. and Howard, J. (1995). The big five quickstart: An introduction to the five-factor model of personality for human resource professionals. `http://www.centacs.com/quickstart.htm`.

[Obernesser, 2003] Obernesser, C. (2003). Cartes cognitives pour la modélisation comportementale. Mémoire DEA, Université de Bordeaux 2.

[Ortony et al., 1988] Ortony, A., Clore, G., and Collins, A. (1988). *The cognitive structure of emotions*. Cambridge University Press, Cambridge, MA.

[Ortony and Turner, 1990] Ortony, A. and Turner, T. (1990). What's basic about basic emotions? *Psychological Review*, Volume 97:p 315–331.

[Picard, 1998] Picard, R. (1998). Affective computing. *The M.I.T. Press*, pages p 194–226.

[Plutchik, 1980] Plutchik, R. (1980). A general psychoevolutionary theory of emotion. *Emotion: Theory, research, and experience*, Vol. 1. Theories of emotion:p 3–33.

[Prat, 2006] Prat, M. (2006). *Processus cognitifs et émotions*. L'Harmattan, Paris.

[Roseman et al., 1990] Roseman, I., Jose, P., and Spindel, M. (1990). Appraisals of emotion-eliciting events: testing a theory of discrete emotions. *Journal of Personality and Social Psychology*, Volume 59(Issue 5):p 899–915.

[Sloman, 2005] Sloman, A. (2005). Do machines, natural or artificial, really need emotions? *NWO Cognition Programme* presentation at Utrecht on the 24th june 2005, available at the following address: `http://www.cs.bham.ac.uk/research/cogaff/talks/cafe-emotions-machines.pdf`.

[Toda, 1982] Toda, M. (1982). *Man, robot and society*. M. Nijhoff Pub., Boston.

[Tong-Tong, 1995] Tong-Tong, J.-R. (1995). *La logique floue*. Ed. Hermes.

[Velàsquez, 1997] Velàsquez, J. (1997). Modeling emotions and other motivations in synthetic agents. In Fourteenth national conference on artificial intelligence and ninth innovative applications of artificial intelligence conference. Menlo Park.

EMOTIONS: THEORETICAL MODELS AND CLINICAL IMPLICATIONS

Sophie Baudic
Inserm U 792
Physiopathologie et Pharmacologie Clinique de la Douleur
Hôpital Ambroise Paré 9 av Charles de Gaulle
92104 Boulogne Cedex
E-mail: baudic@im3.inserm.fr

Gerard H E Duchamp
Laboratoire d'informatique - UMR CNRS 7030
Institut Galilée - Université Paris Nord
99 av Jean Baptiste Clément
93430 Villetaneuse
E-mail: ghed@lipn.univ-paris13.fr

KEYWORDS

Psychology, Biology, Neural network.

ABSTRACT

Research on how the brain processes emotions provides rich and abounding results. The confrontation between theory and clinical practice contributes to improve our knowledge on emotions. In the present topic, theoretical models both cognitive and biological are briefly reviewed in the first part. Interactions between research and clinical practice are examined across two separate approaches in the second part. The first, neuropsychological approach considers Alzheimer's disease to further explore relationships between emotions and memory disturbance. The second considers the anxiety disorder to explore cognitive and behaviour theories of fear. Finally, we address therapeutic applications for each approach which directly come from the theory.

INTRODUCTION

The aim of this topic is to show that researchers and clinicians can interact together rather than exist side by side without any real contact. This interaction is essential for the evolution of the discipline and for patients. Theoretical models provide insights on the structural and functional organization of normal cognition and therefore their evolution is crucial for the management of patients. The therapeutic actions, the strategies of rehabilitation and the development of new tools are based on theory. The present topic depicts in the first part the different theoretical models both cognitive and biological and it considers in the second part two approaches which illustrate the implication of theory in the management of patients. The first is that of cognitive neuropsychology which analyses the dysfunctions caused by Alzheimer's disease to emotions and the second is that of cognitive and behaviour theories that provide therapeutic strategies in the management of panic disorder that is a good model for studying fear in humans.

THEORETICAL MODELS

The study of emotions is hampered by several conceptual problems. The major one is the relationship between emotions and cognition which remains very controversial in the debate between biological (Zajonc 1980) and cognitive (Lazarus 1982) theorists of emotions. The former maintain that emotions and cognition must be considered as two independent systems whereas the latter argue that cognition plays an integral role in emotion (for overview see Schorr 2001).

Before presenting the different theories of emotions, it is essential to define them, because there is a tendency to lump together a number of different affective phenomena under the term of emotion.

1.1 - Definition: How to define emotions precisely?

The exact nature of emotions remains controversial. Disagreements mainly stem from the fact that the sets of phenomena taken into account are very different. Some focus their attention on the simplest aspects of emotions as they appear in animals or in the early stages of the human development. Others are attracted by the complexity of emotional phenomena.

Emotions have two dimensions: expressive and cognitive. The expressive dimension concerns the production of facial emotions or expression of internal states of the brain whereas the cognitive dimension is related to the comprehension of facial emotions expressions and intentions to act. Emotions are feelings that accompany emergent "states of being" like happiness or despair.

According to Ekman (1984), emotions are reactions that last some seconds and must be distinguished on one side by very brief responses (such as the reflex reactions or basic survival-related appetitive behaviours) and, on the other side, by long-lasting affective schemata, such as affects or personality traits.

According to Scherer (1984), emotions are different from a simple state of the organism (behaviourists). They are processes, i.e. a dynamic sequence of different variables whose components are the following: cognitive evaluation of stimuli or situation, physiological activation, motor and facial expression, action draft or planning of behaviour and feelings (subjective states). In accordance with this point of view, function of emotions is adaptive because it allows a large flexibility of behaviour thanks to an elaborate treatment of complex informations.

Examples of emotions are happiness, sadness, anger, fear, surprise and disgust but also despair, pride...

The next part of the present topic provides an overview of major theories of emotions. It resumes and combines the several authors' works: Gainotti (2001), Scherer and Peper (2001). Presently, we shall describe the cognitive theory of emotions.

Psychological theories of emotions

According to the cognitive theory, there are two major models; the componential and the hierarchical. See Scherer and Peper (2001) for an excellent review on emotion theories.

Componential models

Emotions have qualitatively different facets. According to Scherer and Peper (2001), most of the psychological theories postulate that subjective experience, peripheral physiological responses and motor expression are major components of the emotion construct. These three components have often been called the *emotional response trial*. A few theorists include two other components: cognitive and motivational in the emotion process. Componential models differ strongly with respect to the relative role assigned or the amount of attention paid to these different components. Systematic researches in this field are associated with Lazarus (1982) and Scherer (1984).

Hierarchical models

Within this classification, we distinguish "structural from developmental" models. The *structural models* maintain that emotions are hierarchically organized with numerous discrete emotions at the basic level and very few emotional dimensions at a higher level. The basic emotions such as happiness, sadness or fear are viewed as the building blocks of the emotion system while the dimensions of valence, (pleasantness/unpleasantness) and arousal (rest or activation) take place on a higher level in the hierarchy. Emotions at the basic level have an important adaptive function and are devoted to the detection of stimuli that are crucial for the organism's well-being. According to the partisans of these models, complex emotions of higher level derive from the basic ones because of increasing interactions between the emotional and cognitive systems. *The developmental models* are based on the activity of three functional levels: the sensorimotor, the schematic and the conceptual. The *sensorimotor level* is at the first grade of action or interpersonal communication, the basic interactive schemata of human species. It involves a set of expressive-motor programs that are innate and universal. The *schematic level* which is the second stage of emotional processing includes "emotional schemata". The latter are different for each individual as they spring from the association of the basic emotions of the sensorimotor level and the situations of individual experience. The schematic level corresponds to spontaneous and true emotions. The *conceptual level* is based on mechanisms of consciousness and long term memory. It stores the abstract notions such as "what are emotions?" or "which situations provoke them?" or "how to deal appropriately with them according to the social rules?" (Gainotti 2001).

Emotional as a biologically adaptive system

The biological models like their cognitive counterparts have either componential or hierarchical organization.

Componential models

Emotions are set up by several components which are subserved by different anatomical structures. With regard to motor expression, the most commonly accepted view consists in assuming that the hypothalamus is involved in the more elementary level, i.e. in the generation of the autonomic reaction. In neuropsychology, the componential models prompted two main lines of research. The first consisted of studying the correlations between disorders that affect different components of the emotional processing system. The second line of research attempted to clarify the relationships that exist between specific components of emotions and well-defined structures of the brain.

Hierarchical models

The biological theories are in line with a phylogenetic perspective. According to most authors, models with a hierarchical organization are based on the complexity of emotional computation performed by different brain structures with control of the highest functional levels over the lowest ones. Brains structures subserving emotions may be based on complexity of operations performed at different levels. The highest brain structures inhibit, modulate and extend (rather than replace) the earliest functional systems (Gainotti 2001).

The most influential of these models was proposed by Papez (1937) who tried to specify the role played by different cortical and subcortical structures in various aspects of emotional processing. Emotions were subserved by anatomical circuit that is beginning and ending in the hippocampal formation. It includes hippocampus, fornix, hypothalamus, anterior nuclei of the thalamus, cingulated cortex and their interconnections. This description which is not completely erroneous is reconsidered today. New technologies showed that other structures were involved in the Papez's circuit such as the amygdala and the prefrontal cortex. LeDoux and colleagues (1984) provided anatomical and experimental support to the Papez' dual route model. They show that the amaygdala (and not the hypothalamus) is the structure where information coming from the outside world acquires emotional signification.

Another example of complex model was proposed by Gray and McNaughton (1996). They identified specific brain systems: the behavioral inhibition system (BIS), the behavioral approach system (BAS), and the fight-flight system as substrates of these emotional dimensions. The BAS is supposed to regulate approach behavior, sensitivity to reward stimuli and active avoidance behaviour. The BIS is supposed to inhibit instrumental and unconditioned behavior and to control orienting reactions. Structures of different phylogenetic level might mobilize in situations of different complexity the same defensive fight-flight attitude, i.e. when the source of danger is very close and there is no time for analysis or when the situation involves more distant threats and there is more time for analysis. Furthermore, the amygdala might mobilize defensive behaviour in light of potential (rather than actual) events (see for reviews Gainotti 2001, Scherer and Peper 2001).

Each of the models presented above try to capture and explain emotion either as a basic or complex process. These models progress consistently over time and become more efficient. For a long time, emotions were considered as useless manifestations, irrational and a source of interference. So, the first cognitivists tried to eliminate emotional dimensions from their models. The former theorists also considered that emotions were undifferentiated. Indeed, it seems that each emotion corresponds to a distinct functional cerebral unit. Two basic emotions, fear and pleasure, were extensively examined over the past years and the studies showed they involve different circuits. The amygdala is involved in the circuit of fear while the accumbens is necessary for pleasure. However, functional neuroimagery studies show that the human amygdala can differentially respond to changes in magnitude of positive or negative reinforcement conveyed by lexical stimuli (Zalla et al. 2000). Finally, the cognitive models did not take into account the biological constraints provided by the anatomical organization of the brain or the fact that the adaptive systems subserved by this organ have undergone important reorganization during its phylogenetic history (Tucker et al. 2000). In the same way, the biological models of emotions reveal some limits. Most of the neuroanatomic models are based upon data obtained in animals. Divergent neural connectivity subsists across species. Furthermore, some controversies exist between the studies in animals and a generalization to humans is sometimes problematic.

CLINICAL IMPLICATIONS

Neuropsychology of emotions can be considered to be a very young and fresh field of inquiry. The first series of studies conducted in this area have been almost exclusively devoted to the problem of hemispheric asymmetries in representation and control of emotions (Borod 1993). More recently, the focus of attention has been directed to a much boarder array of problems.

In neuropsychology research (organic diseases)

Alzheimer's disease (AD) is a good model for studying the alteration of emotional disturbance because it involves the amygdala which plays an important role in emotions as demonstrated by functional neuroimaging studies (Cahill et al. 1996; Canli et al. 2000).

In the present topic, we shall focus our attention on the relationships between emotions and memory disturbance in AD. When normal controls learn new information, they associate emotional items with additional semantic information or with autobiographical experiences. Furthermore, emotions may serve as a retrieval cue. A person may initially remember how they felt about an event, and that cue may then allow them to generate additional features about the event. To begin with, it is essential to recall the kind of memory impairment involved in Alzheimer's disease (AD). The disease results in significant atrophy of the medial temporal lobe that leads to a dramatic memory deficit. At the early stage, impairment concerns mainly episodic memory (i.e. it refers to knowledge of episodes and facts that can be consciously recalled and related) which is characterized by an inability to learn new information or to recall previously learned information. The decline of cognitive functioning is gradual, involving other impairments as the disease progresses.

Emotional influence on recall was studied by means of cognitive neuropsychology of memory (Tulving et al. 1972). This model postulates that memory can be considered in terms of dissociable systems, distinct processes, and neuroanatomical structures. Within long-term memory systems, episodic memory is typically severely impaired in early-stage of the disease. Semantic memory that underlies knowledge and language is less likely to be significantly affected, although impairments may be observed in some individuals. Procedural memory (i.e. the abilities to gradually acquire and retain motor, perceptual and cognitive skills) is preserved, as are some aspects of priming. Memory can also be considered in terms of the processes of encoding, storage and retrieval (Tulving and Thomson 1973).

Some studies (Kazui et al. 2000; Moayeri et al. 2000) showed that recall of AD patients is typically better for emotional than for neutral stimuli. Memory is also better for neutral stimuli embedded in an emotional context. Other studies did not show such results. They concluded that AD disrupts memory enhancement for verbal emotional information (Hamann et al. 2000; Kensinger et al. 2004). AD patients also demonstrated impairments in emotionally mediated implicit memory (Padovan et al. 2002). Differences across studies are related to the heterogeneity of patient populations (disease severity), difference in the stimuli or the extent of amygdala atrophy (Mori et al. 1999).

Emotional arousal improves episodic memory in patients with AD and gives a clue to the management of people with dementia (Kazui et al. 2000). Rehabilitation of emotions is based on aspects of emotional communication such as prosody. A series of experiments (Thaut et al. 2005) begin to investigate the effect of music as a mnemonic device on learning and memory. More researches are needed to develop a useful strategy for memory improvement.

In cognitive and behaviour therapy (functional disorders)

Panic disorder is a good model for studying fear which is a basic emotion. It can be depicted as a profound blast of anxious affect. The physical symptoms are multiple: shortness of breath, rapid heart rate, dizziness, tingling, sweating. The cognitive symptoms involve automatic thoughts and mental images which tend to be catastrophic, i.e. there is a tendency to exaggerate the dangerousness of a situation and simultaneously to underestimate the control over the danger.

Therapeutic actions that involve cognitive-behaviour therapy are based on classical conditioning theories (Pavlov 1928). Fear conditioning occurs when initially innocuous conditioned stimulus (CS) is associated with an aversive unconditioned stimulus (US) that activates unconditioned fear responses (URs). The CS comes to elicit various conditioned responses (CRs) that share similar characteristics to innate fear responses (Kim and Jung 2006). The best known example of fear conditioning reported by the authors is the little Albert case (Watson and Rayner 1920). Albert was an infant who initially exhibited curiosity over a white rat by touching and playing with it. As Albert's hand touched the rat, the experimenters triggered a big noise behind his head (US) causing him to startle and cry (UR).

Afterwards, when the rat (CS) was placed near Albert's hand, he withdrew his hand and began to cry (CR). This exhibition of fear towards the rat was generalized to other white furry animals and objects.

Treatment of patients suffering from panic disorder involves exposure to fear cues (behaviour therapy) and cognitive restructuring (cognitive therapy). One powerful means of reducing anxiety problems is believed to counter avoidance. Avoidance reduces anxiety in short term, but makes for more anxiety in the long term as avoidance increases over time. *Exposure* involves placing someone in the avoided situation until the anxiety decreases completely. The disappearance of anxiety is called *Extinction. Cognitive restructuring* is used to identify and counter fear of bodily sensations. Patients are encouraged to consider the evidence and think of alternative possible outcomes following the experience of bodily cues.

LeDoux's model (1986) provides a theoretical framework for therapeutic actions of cognitive-behavioural therapy as it establishes a relationship between emotions and cognitive factors. The model postulates the contribution of the thalamus and the amygdala in fear conditioning and anxious reactions. These structures form a circuit that involves immediate survival responses (flight or fight). The connection between the thalamus and the amygdala is the most direct and therefore the fastest. In parallel, another pathway exists which includes the prefrontal cortex and the hippocampus in addition to the thalamus and the amygdala. The prefrontal cortex is involved in more complex conditioning that requiring planning actions. An individual takes more time for cognition to shift from reaction to action and he is seen as an emotional actor who copes with a cognitive plan of voluntary action rather than just a reactor to an involuntarily elicited emotional reaction. In the light of this model, it is easy to understand the therapeutic actions of exposure to fear cues and cognitive restructuring. Exposure to fear cues seems to be mediated by the more direct circuit while cognitive restructuring is supported by the second pathway which is more rational. Extinction of anxiety is explained by the action of prefrontal cortex. Bodily feedback is also taken into account in this model. Somatic responses have an impact on the conscious awareness of emotions. Patients who suffer from panic disorder misinterpret bodily feedback and tend to develop avoidance behaviour.

CONCLUSION

In conclusion, theories provide a better understanding of brain functioning. They produced important results in the field of cognitive neuropsychology. This comprehension leads to profound changes in clinical practice in the evaluation of patients and produces new orientations in the rehabilitation. During the past years, neuropsychologists progressively changed their conception of the brain that is more theoretically and methodologically constructed. Inversely, a single patient's deficit gave rise to new theories or had a part in the improvement of existing models. The single case studies are used to formulate hypotheses about brain processes and thus its role in the historical development of the discipline is crucial. In the case of panic disorders, theory gives supports for understanding the therapeutic actions and leads to new perspectives of treatment in the management of patients. The exchange between theory and clinical practice allows progressive adjustment between the knowledge on the brain functioning and the management of patients.

REFERENCES

Borod, J. 1993. "Cerebral mechanisms underlying facial, prosodic, and lexical emotional expression: a review of neuropsychological studies and methodological issues." *Neuropsychology,* 12, 2493-2503.

Cahill, L.; R.J. Haier; J. Fallon; M.T. Alkire; C. Tang; D. Keator; J. Wu; and J.L. McGaugh. 1996. "Amygdala activity at encoding correlated with long-term, free recall of emotional information". *Processings of the National Academy of Sciences USA*, 93, 8016-21.

Canli, T.; Z. Zhao; J. Brewer; J.D. Gabrieli; and L. Cahill. 2000. "Event-related activation in the human amygdala associates with later memory for individual emotional experience". *Journal of Neuroscience*, 20, 1-5.

Ekman, P. 1984. "Expression and the nature of emotion". In *Approaches to Emotion,* K.R. Scherer and P. Ekman (Eds.). Hillsdale, NJ: Erlbaum, 319-344.

Gainotti, G. 2001. "Emotions as a biologically adaptive system: an introduction" In *Emotional behaviour and its disorders*, F. Boller and J. Grafman (Eds.). Handbook of Neuropsychology, Elsevier, Amsterdam, 1-15.

Gray J.A. and N. McNaughton. 1996. "The neuropsychology of anxiety: reprise." In Nebraska Symposium on Motivation: Perspectives on Anxiety, Panic and Fear, D.A. Hope (Ed.). Lincoln, NE: University of Nebraska Press, 61-134.

Hamann, S.B.; E.S. Monarch; and F.C. Goldstein. 2000. "Memory enhancement for emotional stimuli is impaired in early Alzheimer's disease*." Neuropsychology*, 14, 82-92

Kazui, H; E. Mori; M. Hashimoto; N. Hirono; T. Imamura; S. Tanimukai; T. Hanihara; and L. Cahill. 2000. "Impact of emotion on memory. Controlled study of the influence of emotionally charged material on declarative memory in Alzheimer's disease. *British Journal of Psychiatry*, 177, 343-7.

Kensinger, E.A.; B. Brierley; N. Medford; J.H. Growdon; and S. Corkin. 2004. "Effects of Alzheimer disease on memory for verbal emotional information". *Neuropsychologia*, 42, 791-800.

Kim, J.J.; and M.W. Jung. 2006. "Neural circuits and mechanisms involved in Pavlovian fear conditioning: a critical review". *Neuroscience and Biobehavioral Reviews*, 30, 188-202.

Lazarus, R.S. 1982. "Thoughts on relations between emotion and cognition." *American Psychologist,* 37, 1014-1019.

LeDoux, J.E.; A. Sakagachi; and D.J. Reis. 1984. "Subcortical efferent projections of the medial geniculate nucleus mediate emotional response conditioned by acoustic stimuli." *Journal of Neuroscience* 4, 683-689.

LeDoux, J.E. 1986. "Cognitive – emotional interactions in the brain". *Cognition and Emotion,* 3, 267-289.

Moayeri, S.E.; L. Cahill. Y. Jin; and S.G. Potkin. 2000. "Relative sparing of emotionally influenced memory in Alzheimer's disease". *Neuroreport*, 11, 653-5.

Mori, E.; M. Ikeda; N. Hirono; H. Kitagaki; T. Imamura; and T. Shimomura. 1999. "Amygdalar volume and emotional memory in Alzheimer's disease". 156, 216-22.

Padovan, C.; R. Versace; C. Thomas-Anterion; and B. Laurent. 2002. "Evidence for a selective deficit in automatic activation of positive information in patients with Alzheimer's disease in an affective priming paradigm". *Neuropsychologia*, 40, 335-9.

Papez, J.W. 1937. "A proposed mechanism of emotion." *Archives of Neurology and Psychiatry*, 79, 217-224.

Pavolv, I.P. 1928. "Lectures on conditoned reflexes" New York International.

Scherer, K.R. 1984. “On the nature and function of emotion. A component process”. In In *Approaches to Emotion,* K.R. Scherer and P. Ekman (Eds.). Hillsdale, NJ: Erlbaum, 293-318.

Scherer, K.R. and M; Peper. 2001. “Pscyhological theories of emotion and neuropsychology research” In *Emotional behaviour and its disorders*, F. Boller and J. Grafman (Eds.). Handbook of Neuropsychology, Elsevier, Amsterdam, 17-48.

Schorr, A. 2001. “Appraisal – the evolution of an idea”. In *Appraisal Processes in Emotion: Theory, Methods, Reseach.* K.R. Scherer, A. Schorr and T. Johnstone (Eds.). Oxford: Oxford Univeristy Press, 20-34.

Thaut, M.H.; D.A. Peterson; and G.C. McIntosh. 2005. “Temporal entrainment of cognitive functions: musical mnemonics induce brain plasticity and oscillatory synchrony in neural networks underlying memory” *Annals of the New York Academy of Sciences,* 1060, 243-54

Tucker, D.M.; D. Derryberry; and P. Luu. 2000. “Anatomy and physiology of human emotion: vertical integration of brainstem, limbic and cortical systems.” In *The neuropsychology of emotion*, J.C. Borod (Ed.). New York, Oxford University Press, 56-79.

Tulving, E. 1972. “Episodic and semantic memory”. In: *Organisation of memory,* E. Tulving and W. Donaldson (Eds.). Academic Press, New York, 381-403.

Tulving, E.; and D.M. Thomson. 1973. “Encoding specificity and retrieval processes in episodic memory”. *Psychological Review*, 80, 352-373.

Watson, J.B. and R.R. Rayner. 1920. “Conditioned emotional reaction” *Journal of Experimental Psychology*, 3, 1-14.

Zajonc, R.B. 1980. “Felleng and thinking: preferences need no inferences”. *American Psychologist,* 2, 151-176.

Zalla, T.; E. Koechlin; P. Pietrini; G. Basso; P. Aquino; A. Sirigu; and J. Grafman. 2000. “Differential amygdala responses to winning and losing: a functional magnetic resonance imaging study in humans” *European Journal of Neuroscience*, 12, 1764-70.

NATURAL ECOSYSTEM MODELLING

Detection and reification of emerging systems in populations dynamic simulations using interaction networks and genetic algorithms : a way to exploit Individual-Based Models

Guillaume Prévost and Cyrille Bertelle
LIH - Université du Havre
25 rue Ph. Lebon - BP 540
76058 Le Havre Cedex - France
E-mail : guillaume.prevost@univ-lehavre.fr

Keywords

Complex systems, interaction networks, emerging systems, non-linear differential systems, genetic algorithms.

ABSTRACT

In this paper, we present an hybrid ecosystem modeling based on emerging computation from interaction networks. Initially based on an individual-based modeling (IBM) simulation, we proposed an automatic computation to detect predator-preys systems. After their detection, these systems are replaced by a differential system, during the simulation. In that way, we can change the description level and improve both the computation time and the whole system analysis by detecting some emergent organizations. The description modification between IBM representation to differential one, needs to identify the global coefficients of these differential equations. Because of the complex relations between these two kinds of representations, a genetic algorithm is proposed to solve this identification.

1 COMPLEXITY AS A MATTER OF INTERACTIONS

1.1 Emergence of complex systems

In 1968, L. Von Bertalanffy (Bertalanffy, 1968) describes systems as a set of elements in interaction. Thus, the becoming of an element is strongly linked with its local environment. Eventually, some elements in interaction become related in dynamic and therefore constitute a global entity at a higher level. Finally, entities evolve at their own level and modify their parts.

Recently, most scientists describe the complexity as a result of interactions between elements of a system. They point out the fact that those systems are non-deterministic ones (Cardon, 2003). That fact implies that complex systems can't be study by dividing them into smaller parts but as a self-organizing systems (Cardon, 2005) in which smaller systems sometimes emerge. Those systems stand for local sets of elements temporarily interrelated in dynamic which can aggregate or reject elements (or totally disappear) as they undergo new interactions from their local environment. The General System Theory sketches the basis of the complex systems structure.

As a consequence, we can describe complex systems as a system according to L. Von Bertallanffy. The structure of that system evolves depending of the local interactions between its parts on different scales of time and space. That structure is made of entities (or emerging systems) appearing and disappearing depending of the configuration of the interactions network of its elements. To conclude, the structure of a complex systems results of the feedback between its parts on their own scale and between the scales. A complex system is made of complex systems, the whole influencing the parts and the parts modifying the whole.

1.2 Modeling: an interaction network based approach

As we previously describe, complex systems embody the rise and the disappearing of systems. Since many years, models are being developed to study the evolution of those systems. In ecology and especially in population dynamic, a great diversity of tools coming from mathematics, computer science or medicine for example are used to describe the interrelated becoming of populations in their environment. Each model can be analyzed by considering the associated interaction network between its elements (Prévost, 2005).

Let's us give a simple example with a very classical model: the logarithmic growth of a population. That model only embodies the dynamic of one population limited by an external factor that can be a critical resource. Therefore, the interaction network is simply made of the population and its resource linked by the consumption interaction. As the resource is too low, the consumption is lowering and the growth is weakening. More complicated models induce more complicated interaction networks but it is important to underline the fact that, in particular conditions, one can associate a model to a network.

Some models directly map their associated interaction network in their forms. For example, linear or non linear differential model (using one or more equations) mostly express the interactions as a term of one of their equations. Matrix models (as Leslie ones) fix the interactions between the populations in their values.

1.3 Using Individual based models to study interactions networks

Since its definition in (Angelis and Gross, 1992), Individual Based Models (IBM's) has been widely used in many sciences. Questions remain regarding the use of the results (Grim, 1999). Basically, IBM's differ from global models as they focus on individuals and not directly on populations. Thus, both of them is relevant at a different scale and apply to different kind of studies. IBM's are not only a discrete expression of global models but could induce non-deterministic aspects as individuals are in a local context. We can assimilate individuals in IBM's as elements in our ecosystem[1].

The interactions modeled in IBM's are on a lower scale than interactions in classical models. Despite that fact, if one considers the interaction network between the individuals, one can underline the fact that this interaction network corresponds to a more global one. In complex systems, systems keep on emerging and disappearing. Those systems correspond to temporary global entities and therefore can be modeled thanks to a global model. As we will introduce, we can detect in IBM's temporary stable interactions networks corresponding to global models.

2 MODELING THE DYNAMICS OF POPULATION AT VARIOUS SCALES

2.1 Ecosystem meta-modeling.

We propose a multi-scale model of ecosystems and its application to population dynamic. That model is based on the General System Theory. It also consider ecosystems as open systems meaning they exchange fluxes with their neighbors (Frontier, 1999). Moreover, it describes ecosystems as holarchic systems (Koestler and Smythies, 1969). Thus, they are organized in many levels, each level influence the level on top of it and the level under it. Finally, it divides ecosystems in compartments. Each compartment has special environmental conditions and is linked with other compartments. That abstract model explains the organization and the interactions between the scales and parts of an ecosystem.

2.2 Application to population dynamics.

Let's use that model to study population dynamics in relation with the environment conditions. First, elements are individuals form different species that can be modeled thanks to individual based models (including super individuals) on a very local scale or according to population models at a global level. So the model has two levels: a local one with localized individuals (as discrete sets of states) and a global one with populations (continuous values). As we introduce environment as a key factor of our model, we should add it to the elements of the model. Therefore, global values as states of our compartments should represent the environment conditions which associated model are differential ones[2] in addition with its interactions with individuals and populations. Finally, we should describe the interactions between individuals and environment conditions. To achieve that, we separate the species in three categories: consumers (eating preys to sustain their organic matter and depending on environment conditions to survive), producers (producing and consuming environment conditions) and detrivors (consuming dead organic mass and producing environment conditions, mostly mineral salt). Of course, others feedbacks between populations and environment are introduced.

Using that model, we can point out drastic feedbacks induced by environment conditions between populations as well as the influence of space on population dynamics.

2.3 Self regulation of predator-prey systems induced by interactions with environment conditions studied with IBM's.

A generic distributed platform called H2O corresponding to the basic model and using active objects to simulate global entities and reactive agents for local elements has been developed. Special behaviors has been developed for the platform's objects to fit the model of our populations and individuals. The aim is to point out the particular population's dynamics induce by the interaction with the environment meaning environment conditions constitute an indirect interaction between populations.

Let's consider two populations :

1. One made of planktons responsible for the photosynthesis and therefore the production of O_2 depending on

[1] Of course, they are not the only elements.

[2] if the environment condition is not static

light intensity. As they do the photosynthesis, planktons raise their organic matter until they can reproduce.

2. One made of fishes consuming O_2 and eating planktons to survive and reproduce.

Those populations are living in a compartment with a particular light intensity in which O_2 is a environment condition. We should associate a model with each part of the study:

- Fishes are modeled as super-individuals. Each step of time, a death rate is applied on their number. They also eat preys in their radius of action (depending on their number) which raise their organic matter. Eventually, they could reproduce if their organic matter is high enough. If no preys are near, they move until they find one. Finally, they breathe and undergo a loss of number if they cannot.

- Planktons are modeled as super-individuals. Depending on their number and on the light intensity, they produce O_2 and raise their number.

- The compartment has O_2 and light as environment conditions. Light do not vary. O_2 increase periodically due to external supply. The compartment's space is a bidimentional grid.

- fishes predation rate is calibrated to be far too high compared to planktons reproduction rate.

First, we deprive the fishes of breathing meaning they don't need to consume O_2 anymore. Using that partial model, we obtain a well-known case of populations dynamics of type 1. Predator's number undergo a period of increase until prey's number reaches nil. Then, their number falls to nil too as they don't have their resource anymore. Preys disappear as predation is too hard. The time for plankton to be annihilated lay on the efficiency of predator to find them all in the space of the simulation.

Secondly, we fix that fishes should breath to survive. Many simulations show us that predator's number and prey's numbers evolve in phase. First, prey's number decrease due to predation as fishes one increase. As preys became too rare and cannot produce enough O_2 for fishes, fishes population decrease until prey's are able to multiply again and cope with the air production. Thus, fishes number can go up again. The previous phenomena repeats. Eventually, space makes the two population not to encounter. It ends with an excess of O_2 making fishes to eat all planktons without suffering from lake of O_2. A example of simulation's trace is shown in figure 2

To conclude, we sketch how environment conditions constitute a drastic factor of indirect interactions between populations dynamic and how the model helps to take them into account at several levels.

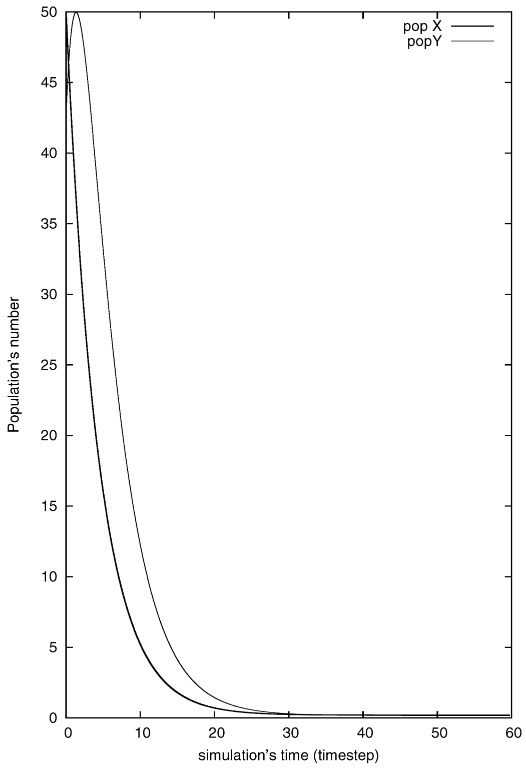


Figure 1: Simulation's curves without breathing activity

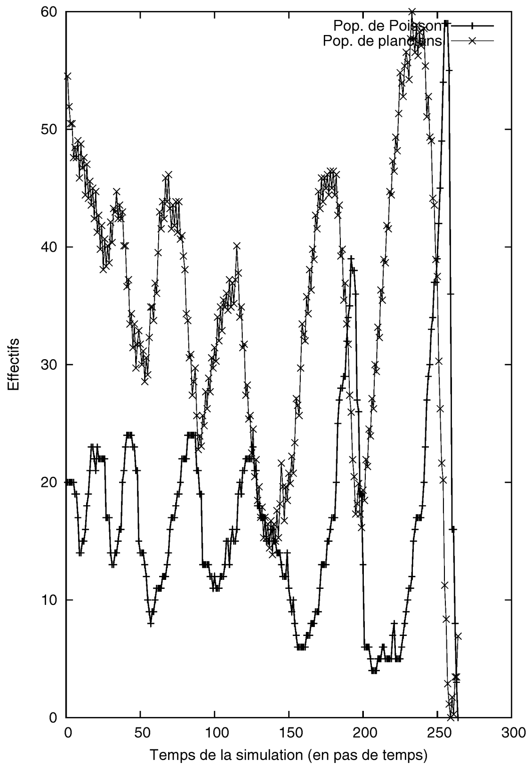


Figure 2: Simulation's curves with breathing activity

3 INTERACTION-BASED APPROACH TO DETECT SYSTEMS AND GENETIC ALGORITHM TO FIX THE PARAMETERS

3.1 Analyzing the interaction network of the simulation.

As the simulation runs, agents representing individuals or active objects representing populations interact at their own level and with their environment according to their perception. Eventually, those parts become strongly link in dynamic and interactions remains between them constituting a particular interaction network. As we previously state, we can associate an interaction network to a model. Therefore, the network of an emerging system could correspond to the one of a well-known global model and we can compute the dynamic of that emerging system at a higher level using the corresponding model. Of course, we should be able to define if an emerging system is stable enough to justify a change of scale in our simulation. We simply fix that if a particular interaction network remains stable during a certain period, the emerging system should be reify. Secondly, we should be able to evaluate interactions in order to compare them. To fulfill that task, we consider the influence on the organic mass of the element in interaction with another element induced by the interactions during the significant period. That number is the valuation of the edge between the nodes representing the two elements and pointing to the node representing the predator.

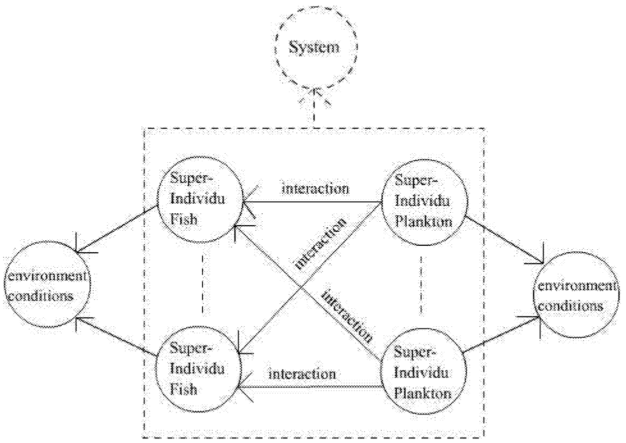
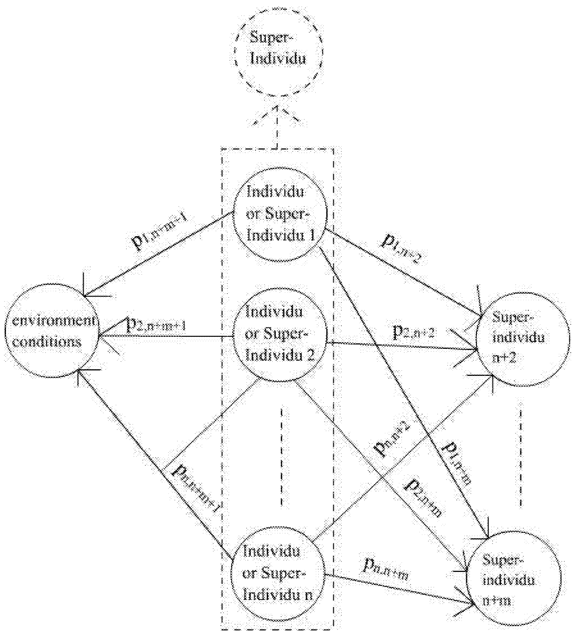


Figure 3: Individuals interaction network corresponding to a Lotka-Volterra system

Figure 3 and 4 show two case of interaction network corresponding to our case of study describe in 2.3. First one simply implies that individuals and super-individuals are in interaction with the same individuals and thus can be consider as a a single super-individual. Second one corresponds to two sets of individuals (preys and predators) that only interact together meaning the set of predator only feed on the set of preys and the set of preys is only consumed by the set of predators. Therefore, they stand for a local system that can be model thanks to a Lotka-Volterra model linking the dynamics of the two populations. By the way, we could extend those interaction networks to the general case as shown in figure 5.



$\forall x1,x2 \in [1,n]$, $\forall$ y $\in$ $[n+1,n+m+1]$, $|p_{x1,y} - p_{x2,y}| <$ Beta

Figure 4: Individuals interaction network corresponding to a super-individual

Thus, the simulation dynamically achieves the making of the interaction network corresponding to the computation of its model on the lower level (IBM's). In parallel, it analyzes that interaction network in order to detect local interactions networks corresponding to a particular global model (see 3 and 4). When a local interaction network remains stable for a sufficient period of time, the simulation gathers its elements in a system corresponding to the associated global model and computes it on the global level.

3.2 From IBM's to global model : calibration with genetic algorithm.

3.2.1 Passing from discrete model to continuous one

IBM's are discrete models that can eventually be issued from game theory. We see how we can pass from a set of individuals on the IBM's level to a global system on the higher level. One of the critical problem is that IBM's and global models can differ in nature. In our example, IBM's are dis-

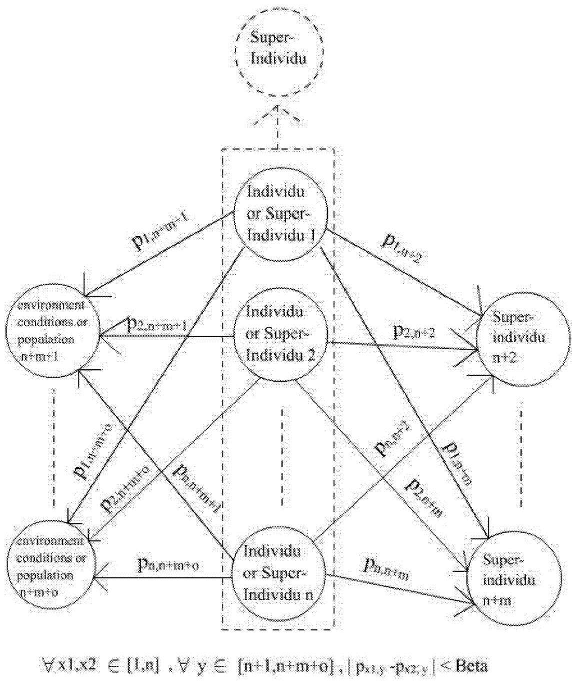

Super-Individu
Individu or Super-Individu 1
Individu or Super-Individu 2
Individu or Super-Individu n
environment conditions or population n+m+1
environment conditions or population n+m+o
Super-individu n+2
Super-individu n+m
$p_{1,n+m+1}$
$p_{2,n+m+1}$
$p_{1,n+2}$
$p_{2,n+2}$
$p_{1,n+m+o}$
$p_{2,n+m+o}$
$p_{n,n+m+1}$
$p_{n,n+2}$
$p_{1,n+m}$
$p_{2,n+m}$
$p_{n,n+m+o}$
$p_{n,n+m}$
∀x1,x2 ∈ [1,n] , ∀ y ∈ [n+1,n+m+o] , | $p_{x1,y}$ -$p_{x2,y}$ | < Beta

Figure 5: General individual interaction network corresponding to a Lotka-Volterra system

crete models whereas global models could be continuous. It is the case when passing from a set of individual to a Lotka-Volterra model. In fact, the number of individuals of the set of super-individuals evolve according to a discrete time but we must recreate the same dynamic with a Lotka-Volterra model.

$$\begin{cases} \frac{dX}{dt} = a.X - b.X.Y \\ \\ \frac{dY}{dt} = -c.Y + d.X.Y \\ with\, a > 0,\ b > 0,\ c > 0,\ d > 0, \\ \ a \leq 1,\ b \leq 1,\ c \leq 1,\ d \leq 1. \\ \end{cases}$$

The population dynamic between time t_0 and t_n of the set of individuals on the IBM's level is a discrete set of X_i, Y_i where $X_i$is the number of preys and Y_i the number of predators with i ∈ $[\![t_0,t_n]\!]$. Therefore, we will try to determinate parameters a,b,c and d for the Lotka-Volterra equations ideally to verify $X(i) = X_i$ and $Y(i) = Y_i$ with i ∈ $[\![t_0,t_n]\!]$. Of course, we will mostly search for solutions nearly matching that criteria using heuristic approach.

3.2.2 The genetic algorithm.

Genetic algorithm belong to meta-heuristic approach for finding approximate solutions to optimization problems. The principle rely on the crossing and mutation of a population of chromosomes representing solutions of our problem. An heuristic is used to evaluate the solutions. Using that fitness, a selection is made to keep and cross over the best solutions in order to generate a new population that will serve in a new cycle of the algorithm. The genetic algorithm go on until a satisfying solution has been found.

Let's consider a population X of preys and a population Y of predators and n+1 points $(X_i,\ Y_i)$ with i ∈ $[\![t_0,t_n]\!]$representing the trace of their numbers coming from their simulation at the individual level from time t_0 to t_n. Our concern is to fix parameters a,b,c and d for the non-linear system describes in 3.2.1 to minimize $X(t_i) - X_i$ and $Y(t_i) - Y_i$.

First, we use chromosomes described in figure 6 and decide of a fitness which simply minimize $X(t_i) - X_i$ and $Y(t_i) - Y_i$. Obviously, the solutions obtained show that the algorithm doesn't favor solutions that match the phase of the discrete set of points. Therefore, it gives indifferently chromosomes that perfectly correspond to the objective for some period but differ in phase and solutions with a correct phase.

To cope with that, we change the fitness for a least square:

$$\sum_{i=0}^{n}(X(t_i) - X_i)^2 + (Y(t_i) - Y_i)^2$$

a	b	c	d

Figure 6: Chromosome used in the algorithm.

That fitness allows us to find solutions that match the phase of the set of point. Another issue has to be take into account. In fact, the space of solution is clearly wide as the algorithm must fix four parameters whose values belong to the continuous interval [0,1]. Thus, the initial population of chromosomes only represents a few part of all the possible combinational solutions. Therefore, we add to the mutation another factor that introduce new genes : migration. That means that every new population's baddest solutions are replaced by new chromosomes. Those arrival corresponds to the mechanism induced by migration between populations.

Large scale tests have been made to check many features. We wanted to know whether the migration mechanism improves the algorithm or not and if the better solution come from a crossover or not. Finally we examine the influence of the number of chromosomes and loops on the efficiency of the algorithm.

We end with the following conclusions:

1. all of the solutions of the algorithm are made by a crossover ;

2. with number of chromosomes and loops fixed, near the totality of the solutions were far better with the modified algorithm (using migration);

3. the greatest the population is, the fastest the algorithm improves its solution and the better it is to avoid local extrema ;

4. the number of loop has an influence on the performance of the algorithm has more loop implies a better solution. On a first time, the improvement is drastic until it reaches critical number of loops where it is hard to get a better solution. So, the curve linking the number of loop and the fitness of the solution looks like a logarithmic model one.

The figure 7 shows the difference in phase between the set of points 9 and one of the approximate solution 8 found thanks to the genetic algorithm.

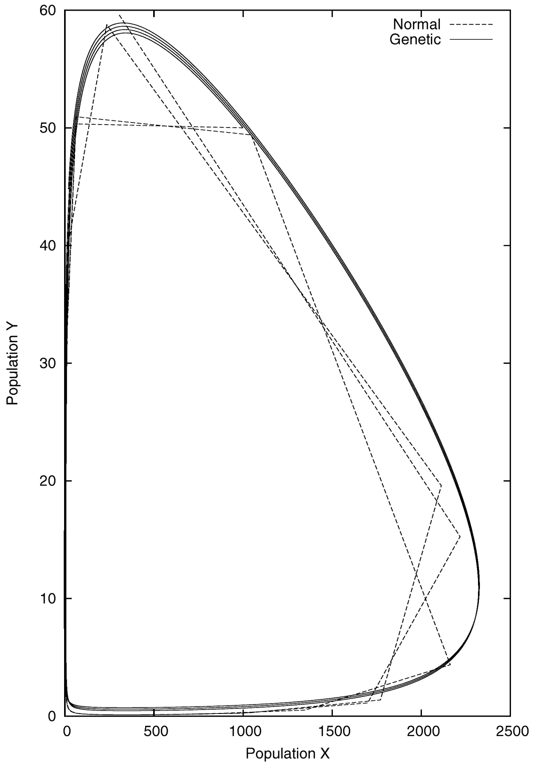


Figure 7: phase diagram comparison between the set of point (labeled normal) and the genetic algorithm solution.

4 CONCLUSION

The present work focuses on studying natural systems and especially the way their complexity expresses in their dynamic. We mainly try to integrate previous work and restore them in the field of complexity. By choosing that approach, we adopt a bottom-up point of view. In fact, we base our

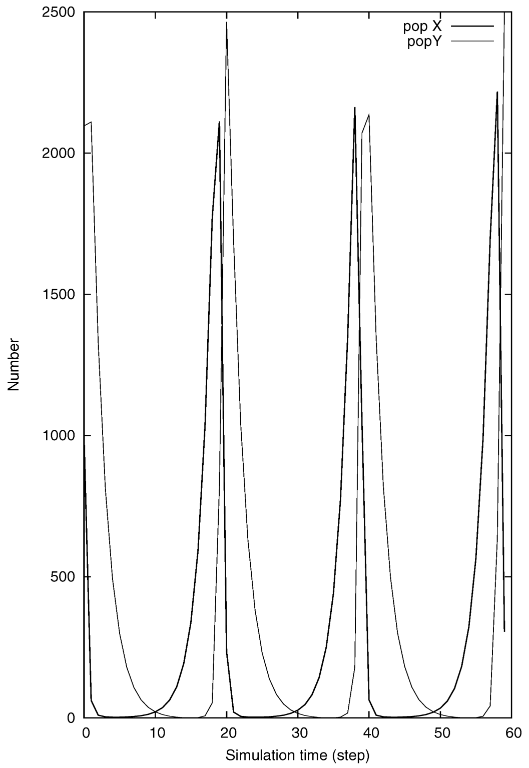


Figure 8: Discrete set of points X,Y plotted with lines.

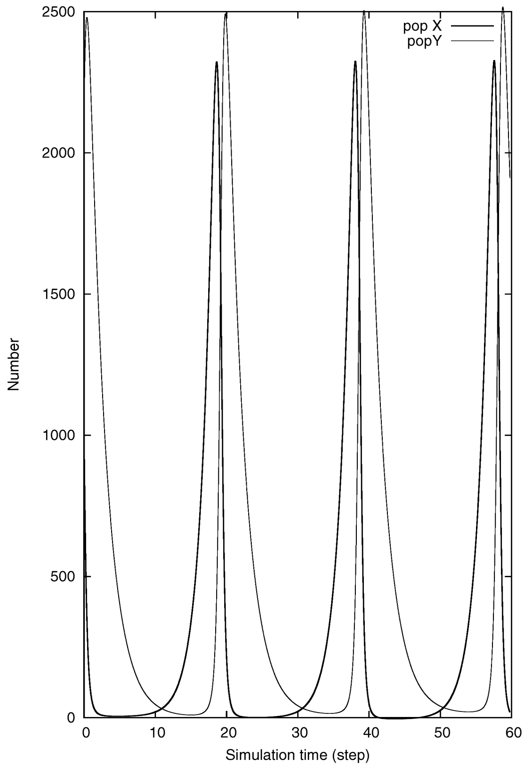


Figure 9: Curves of the approximate solution.

work on the analysis of the becoming of the elements of systems and mainly on their interactions. Investigating population dynamic, we exploit the individual based models results and interrelated those with global modeling to study natural systems structure dynamic and the link between abiotic factors and populations. The end of the paper presents a way to take advantage of previous works in modeling by adopting the interaction-based point of view to appreciate the nature of the systems we detect. As we do so, we are able to give a relevant mathematic expression of the system as we see with the example of the Lotka-Volterra system and the genetic algorithm. That technique should be generalized to other case of systems and different global models. The whole work is a contribution to the analysis of IBM's results in the field of complex systems and constitute a link between global modeling and IBM's.

References

Angelis, D. D. and Gross, L., editors (1992). *Individual-based models and approches in ecology*. Chapman and Hall.

Bertalanffy, L. V. (1968). *General system theory foundations development applications*. George Brazille Inc., New York.

Cardon, A. (2003). *Modéliser et concevoir une machine pensante*. Automates Intelligent's.

Cardon, A. (2005). *La compléxité organisée, systèmes adaptatifs et champ organisationnel*. Hermès-Lavoisier.

Frontier, S. (1999). *Les écosystèmes*. puf.

Grim, V. (1999). Ten years of individual-based modelling in ecology: what have we learned and what could we learn in the furture. *Ecological Modelling*, (115):129–148.

Koestler, A. and Smythies, J. (1969). *Beyond reductionism*. Hutchinson.

Prévost, G. (2005). *Modélisation d'écosystème multiniveaux par des approches mixtes*. PhD thesis, Université du Havre.

Model and simulation engineering in the field of ecology using web ontology and XML

Guillaume Prévost and Cyrille Bertelle
LIH - Université du Havre
25 rue Ph. Lebon - BP 540
76058 Le Havre Cedex - France
E-mail : guillaume.prevost@univ-lehavre.fr

Keywords

Ontology, web, XML, aquatic ecosystems, computation

ABSTRACT

We present in this paper a ecosystem modeling methodology based on ontology description. Meta-models are predefined and specialisations can be given by the users who will describe the specific populations of their case studies. From the data obtained by the ontology using Protege, a XML file is generated. Then a second tool is used to distribute the user model into a simulation, using a specific platform of an hybrid ecosystem simulation developed in Java.

1 NOWADAYS MODELING IN ECOLOGY

Recently, models become frequently used tools in ecology in order to understand or predict the becoming of entities (standing for individuals, populations, settlements or ecosystems for example). At the same time, types of models evolve in their form, complexity and levels of description resulting in an obvious increase of hardness to conceive a model and compute it. Computer science itself gives many way to compute the model. In conclusion, making a model corresponding to a particular case of study, computing it and analyzing the results require many skills and knowledge in very different types of sciences making it difficult to use successfully modeling with large problems. Therefore, model engineering gives clues to help users in the different steps of modeling and computation.

In the present paper, we will present an on-line framework designed to ease the process of model and simulation making in the field of aquatic ecology. That framework can be split in three parts:

1. an ontology of aquatic ecosystem modeling standing for a meta-model ;
2. a website corresponding to an on-line version of the ontology allowing users to apply their case of study to the meta-model and therefore generating an XML file corresponding to a model of their case of study ;
3. a tool using XML files to generate distributed simulations.

The principle of that framework is that each user should find sufficient tools to achieve the process of modeling. In addition, one can bring his own ability by increasing each part of the framework if he wants. Thus, one can complete the knowledge about a particular problem in ecology, add new kind of models or introduce new way to compute models if he has some skills in those sciences.

2 ONTOLOGY OF THE ECOSYSTEM MODELING

2.1 Brief presentation of the model.

We start with a model of aquatic ecosystems described in (Prévost, 2005).

Basically, that model describes ecosystems as complex natural systems according to the General System Theory (Bertalanffy, 1968). Thus, ecosystems are made of elements interacting in their local neighborhood. Those local interactions imply a particular becoming of each elements and particularly the making of emerging entities on a global scale. Entities evolve at their own level and influence their parts. Then, aquatic ecosystems are open which means that they exchange fluxes with other ecosystems. Those fluxes structure the ecosystems as they are crossing it. Finally, one can describe ecosystems structure as an holarchy standing for a structure made of level. In an holarchy, upper and lower levels mutually constrain their way of evolving.

That model explicits the structure of the ecosystem, the feedbacks between its parts and the interactions between the levels. In addition, it distinguishes different levels where some kind of model has to be used. For example, we introduce IBM's at a lower level and link that level with a global one where populations are represented as continuous entities. We should underline the fact that a dynamic level embodying systems exists. It allows us to introduce emerging set of elements (aka systems) in our futures simulations and therefore to study emergence in aquatic ecosystems.

2.2 Ontology as a medium of engineering.

The model itself, despite the fact that it contains implicit informations, can't be consider as an efficient way to communicate the knowledge about how to make a model corresponding to a case of study. In order to do so, we make an ontology of that model which constitutes a meta-model.

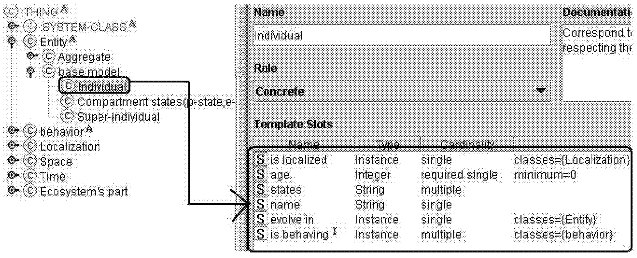
Ontology (Gruber, 1991) stands for clear definition of a domain of knowledge respecting the following criteria :

1. Ontology must be clear and splits the domain of knowledge in many concepts that are related. Concepts are linked and the label of the link describes the way they are in relation. A concept can be a particular case of another. In that case, the latter inherits from the former. Every concept and link must have a clear definition.

2. Ontology must be sharable. In fact, one should be able to use concepts or links for its particular domain of knowledge. For example, basis concepts of ecology are common to different domain of knowledge or one should want to apply a topology of links to its particular domain of knowledge that has nothing to do with ecology.

3. Ontology can be completed meaning users should be able to add new concepts to the ontology for it to fit their needs.

All those features fit with the needs of model engineering so we made an ontology of our model using Protege. That ontology explicits the structure, interactions and feedbacks described in the model as well as its inner way to relate models and levels. Moreover, it is linking the model with concrete elements from ecology helping users to apply the model to their case of study. For example, we introduce a decomposition of populations between producers, decomposers and consumers and link it with the different models of our meta-model. Therefore, we can complete the meta-model's range by increasing the concepts of this part of the ontology.
Finally, ontology are not a static view of a domain of knowledge but should be applied to particular usecase. Doing so applies the knowledge to the particular case and therefore helps us understand it and see how all the parts of this usecase are interrelated. That process is sometimes called instantiation as it consists in giving concrete values from our usecase to all the linked concepts of the ontology.



```
C :THING
  C :SYSTEM-CLASS
  C Entity
    C Aggregate
    C base model
      C Individual
      C Compartment states(p-state,e-
      C Super-Individual
  C behavior
  C Localization
  C Space
  C Time
  C Ecosystem's part
```

Name: Individual

Documentati
Correspond t
respecting th

Role: Concrete

Template Slots

Name	Type	Cardinality	
S is localized	Instance	single	classes={Localization}
S age	Integer	required single	minimum=0
S states	String	multiple	
S name	String	single	
S evolve in	Instance	single	classes={Entity}
S is behaving	Instance	multiple	classes={behavior}

Figure 1: The individual's class in Protégé

2.3 Presentation of the ontology in Protege.

2.3.1 Particularity of Protege.

Protege is a well-known freeware commonly used in computer science to conceive ontology and exploit them. It adopts an object way to represent concepts and links. Therefore, every concepts stand for a class with a clear definition of the concept it is representing. In addition to that definition, particular attributes have to be added by the user. Those attributes are called slots and constitute the links between concepts. Thus, when a concept A is linked with another concept B, a slot is defined in the slot A which type is the class representing concept B. The name of the slot describes the link itself and a clear definition must be added. Additional features of slots exist as the cardinality or the fact that the link is an inverse one.

Finally, making an ontology in protege consists in making a class hierarchy which classes are linked.

2.3.2 The ontology itself.

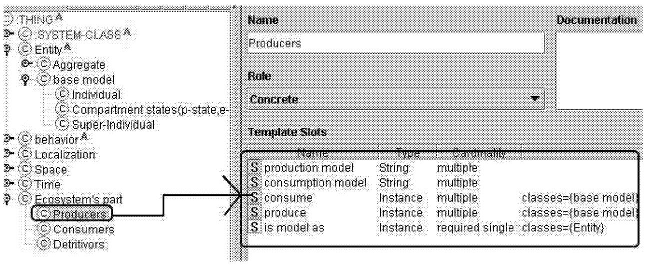
The ontology is divided in four parts representing particular concerns about modeling and whose elements are linked.

Entity stands for every type of data represented in the model. It is named and linked with mathematical values that are the states of the datas (for example, height or number). Therefore, particular subclasses of entity have been defined describing particular way to model datas at particular scale :

1. global entities represent non localized elements described by continuous values on a global scale (for example, settlements).

2. Individual gatherers all way to model discrete entities in a local space. Therefore, it is divided between single individual which represents every individual as a particular entity and super-individual which regroups some individuals that are localized at the same place and have equals states.

3. System represents a set of entities (or its subclasses) which are interrelated in dynamic. Particular models



Name

Producers

Role

Concrete

Template Slots

Name	Type	Cardinality	
S production model	String	multiple	
S consumption model	String	multiple	
S consume	Instance	multiple	classes={base model}
S produce	Instance	multiple	classes={base model}
S is model as	Instance	required single	classes={Entity}

Figure 2: The producer class : subclass of ecological's part.

(as Lotka-Volterra equations or example) exist describing those systems.

4. Compartment stands for a space with particular values describing its abiotic and environmental factors. They embody individuals, global entities and systems.

Those entities are in relation with a behavior standing for the model describing the evolution of its values.

Behaviors are a particular type of classes defining a way to describe the dynamic of an entity. Indeed, it is linking an entity with a particular tool (rule or equation) that will make its values evolve according to the description of space and time. The tools come from mathematic, physics or computer science for example and are classical in modeling. Particular behaviors are associated with the different subclasses of entity and corresponding to their level.

As we sketch, entities could be localized in space. Therefore, different type of **space models** have been defined including continuous or discrete ones, network, grid... **Time models** are also available as a set of classes representing some ways to introduce space in a simulation (event driven for example).

Last but not least, a particular class named **ecological part** 2 fulfills a critical task. In fact, that class and those that inherit of it constitute ecological knowledge linked with the modeling concepts from the other parts. Thus, as a user starts an instantiation of the ontology, he applies his ecological knowledge to that part. Then, the links between that parts and the other parts make the user asking himself questions about how to model his case referring to the definition of the ontology. Therefore, the ontology naturally makes him assuming the task of the modeler as he instantiates it.

2.4 Example of modeling of a coral reef using the ontology.

Let's show a brief example of the modeling of a coral reef with the ontology. The usecase is quite current and can be found in (Alavarez-Hernandez, 2003). It is made of a Mexican coral reef where we want to understand the relationship between the populations. Of course, some populations are preys and others are predators 3.

We shall now demonstrate how the user operates to fulfill the task of modeling. Considering the user has no particular knowledge about modeling, the first classes he will be able to instantiate are ecological part's subclasses. Randomly, we decide to begin with the introduction of "Sharks and scombrids" in the ontology. The "ecological part" gives us the choice between "detrivors", "consumers" and "producers" and brings us to ask ourselves whether those animals are part of one or another category. Once, we decide that "Sharks and scombrids" are consumers, we must fill each slot of that class and therefore decide if we will represent them as individuals, super-individuals, populations...

Those classes are well-defined and thus the choice of individuals appears to be the right one as "Shark and scombrids" are very few in number, large in size and act lonely. Therefore, we shall now complete an instance of class "individual" that is linked with our previous instance of class "consumers". Doing so makes us defining a type of space model to locate the "shark and scombrids". We choose a a model with two coordinates and naturally implies the choice of global space for our simulation restraining us to 2D grid or continuous space for example. Then, we should cope with a slot named "belong to" linking an individual with an environment. That environment represents the coral reef with all the datas about its layout. We already define that it will be represented with a 2D grid so we shall add abiotic factors to that it. The individual class leads us to define behaviors for our "Shark and scombrids". As they are consumers, we must define a predator behavior consisting in linking that behavior with preys of the "Shark and scombrids". That will end with the definition of many other classes. Of course, we should define the function that will model the predation of the "Shark and scombrids" on each one of the population of preys. We simply use here a level that increases each time an individual of specie "Shark and scombrids" eats a prey.

The previous example underlines the fact that the instantiation is a logical process that both leads the user and gives him the information to take his decisions about modeling. Now, we will present an on-line framework designed to share that ontology on a large scale.

3 WEB TOOLS FOR MODELING AND COMPUTING

Actually, Web 2.0 becomes a most spoken of topic highlighting the rise of internet communities that share tools and informations. In the field of that phenomenon, we propose a web version of the ontology that will keep the features of common ontology with ability to be shared among the whole community of potential users (ecologists,

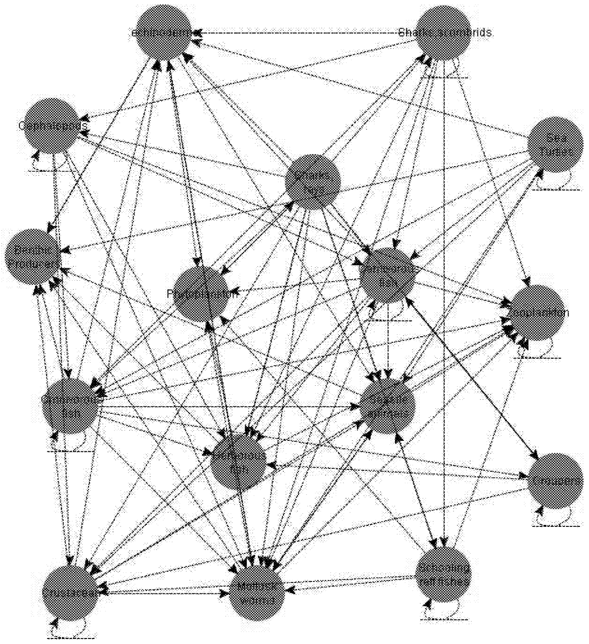


Figure 3: The potential foodweb issued from the instantiation displayed in Protege.

computer scientists ...). In particular, that web ontology should be expandable allowing each person to contribute the knowledge of the whole community. The web tool is made of a website assuming similar taskes to the software Protégé in a relevant form for the internet. Its output corresponds to the model associated to the usecase and is a set of XML files.

As we propose a tool to make a model, we also provide another tool to make a distributed simulation out of it. That tool converts XML files into a code in Java using a simulation library developed simultaneously to the meta-model. Indeed, the ontology is generic so one could use the XML files to make its own simulation on another platform. An interesting point is that the simulation tool should also be increase. In fact, some informations added to the ontology should have to be convert in a computable way (like mathematic models add to behaviors subclasses for example). Thus, the simulation tool will also be a community tool.

As a conclusion, we give access to all the steps of modeling and computing thanks to internet. Those tools are only prototypes for the moment and final version will be released after a significant period of test.

3.1 Website for the ontology

For instant, the website is made thanks to PhP and MySQL. It is made of two parts.

1. The first one is a ontology administration one where the user can consult all the classes already defined in the ontology. Moreover, he can add new slots to each class or create new classes that will be instantiable.
2. The last one is an instantiation one corresponding to the same process in Protégé and ending with the making of XML files for the user.

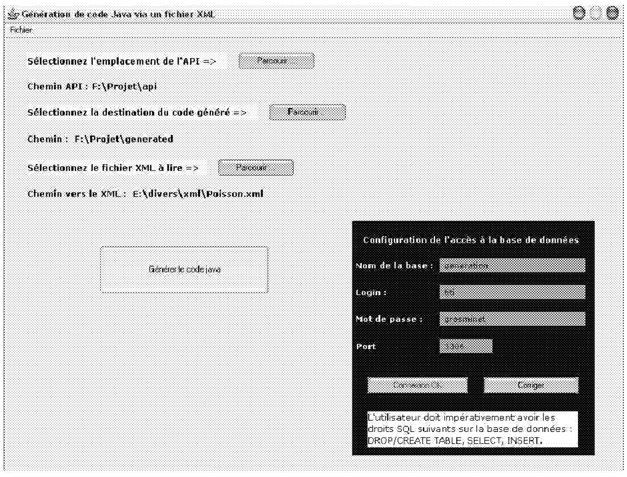
3.1.1 Web-ontology administration issues and instantiation

For instant, the ontology administration one suffers from some issues making it less relevant as a web 2.0 tool. In fact, as a user modifies the ontology, the concepts or links he adds are directly available for all the users of the ontology. That may induce several problems as concepts could be not correct for the knowledge domain, too particular or introduce in a bad shape. As the ontology is a common object, its coherence should be strictly checked. To achieve that, many solutions are possible.

First, we could submit new features to the approval of the whole community which stands for a really uptodate solution. We could also centralize all permissions on an administrator that will accept or deny changes and also provide supports to help users to introduce their features the good way. Obviously, both solutions are depriving users from its own right to use its own version of the ontology. Therefore, we should allow him to use his own version only in the field of its session until the approval for the whole community.

It is important to underline the fact that every information contained in the ontology should be transposed in an XML form for the website to generate the XML files corresponding to the model. The XML informations should respect some rules for it to be used for generating simulations. For example, XML should sum up the informations issued of instantiation in the best way. Definitions are already available in the ontology so only concrete informations defined by the user should be contain in XML tags. Some rules are grammatical rules and won't be exposed anymore. The XML files contain both the ontology's concepts and the values associated to them during the instantiation.

As a web 2.0 tool, the ontology oughts to evolve and grow as users add new concepts. There is already too much concepts so we should face and solve problems of ergonomy. First, we decide to open a window for each instance of a concept. As concepts are linked one with many others, it ends with a process that can't be easily handle by users. So we decide to display instance on the same window. Of course, it induces a huge window but ease the instantiation. To cope with the size of the windows, we will introduce a frame displaying the hierarchy of the instantiation. Thus, the user would easily evolve in the process.



Sélectionnez l'emplacement de l'API =>

Chemin API : F:\Projet\api

Sélectionnez la destination du code généré =>

Chemin : F:\Projet\generated

Sélectionnez le fichier XML à lire =>

Chemin vers le XML : E:\divers\xml\Poisson.xml

Configuration de l'accès à la base de données

Nom de la base : generation

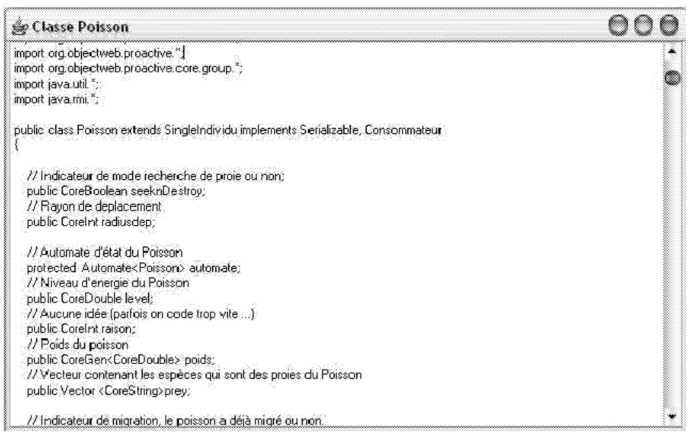
Login : [illegible]

Mot de passe : grosminet

Port 3306

L'utilisateur doit impérativement avoir les droits SQL suivants sur la base de données : DROP/CREATE TABLE, SELECT, INSERT.

Figure 4: The GUI corresponding to the tool.



```
import org.objectweb.proactive.*;
import org.objectweb.proactive.core.group.*;
import java.util.*;
import java.rmi.*;

public class Poisson extends SingleIndividu implements Serializable, Consommateur
{

   // Indicateur de mode recherche de proie ou non;
   public CoreBoolean seeknDestroy;
   // Rayon de deplacement
   public CoreInt radiusdep;

   // Automate d'état du Poisson
   protected Automate<Poisson> automate;
   // Niveau d'energie du Poisson
   public CoreDouble level;
   // Aucune idée (parfois on code trop vite ...)
   public CoreInt raison;
   // Poids du poisson
   public CoreGen<CoreDouble> poids;
   // Vecteur contenant les espèces qui sont des proies du Poisson
   public Vector <CoreString>prey;

   // Indicateur de migration, le poisson a déjà migré ou non.
```

Figure 5: Example of a generated classfile displayed with the GUI

3.1.2 Tool to generate simulations

Simultaneously to the website, we propose a tool that convert XML files into a simulation using our platform. That platform has been well-exposed in (Prévost, 2005). Briefly, it consists in a multi-level API based on active objects. It allows to compute many type of models at different levels using different numerical methods.

The tool is based on a syntax analysis of the XML files. Two types of parsers were available to realize that. First ones were based on the making of lexical trees of the file. The second ones only read and interpret the files. The way we write the files allows us to use the second type of parsers. Therefore, we decide to use SAX, a java-based parser allowing us to take advantage of connectivity between java and database. Partial translations of concepts in XML are already stored in a mySQL database for the website. Therefore, we can use that database to achieve the transcription. It only consists in defining the writing-rules in the parser. The tool is provided as a graphical program 4 that connects over the internet to the database with readonly rights. It contains both the parser and the simulation API. After having run the tool on its XML files, the user obtains a standalone simulation 5.

4 CONCLUSION

The two projects exposed here are parts of an entire framework aiming at ease modeling and simulation's making. Moreover, we take advantage of the Web 2.0 to tackle a main problem in research: knowledge sharing. Those tools would be made as community ones and will induce discussions on modeling and computing research between users. Simultaneously, works from many users will be gathered and will interact. That fact will surely end with a brand new type of work by increasing interactions between research workers on all the sides of modeling. finally, it could be a tool to gather all specialist working on the different issues of modelling and computing.

References

Alavarez-Hernandez, J. (2003). Trophic model of fringing coral reef in southern mexican caribean. *From Mexico to Brazil: Central Atlantic fisheries catch trands and ecosystems models.*

Angelis, D. D. and Gross, L., editors (1992). *Individual-based models and approches in ecology*. Chapman and Hall.

Bertalanffy, L. V. (1968). *General system theory foundations development applications*. George Brazille Inc., New York.

Cardon, A. (2003). *Modéliser et concevoir une machine pensante*. Automates Intelligent's.

Cardon, A. (2005). *La compléxité organisée, systèmes adaptatifs et champ organisationnel*. Hermès-Lavoisier.

Frontier, S. (1999). *Les écosystèmes*. puf.

Grim, V. (1999). Ten years of individual-based modelling in ecology: what have we learned and what could we learn in the furture. *Ecological Modelling*, (115):129–148.

Gruber, T. (1991). The role of common ontology in achieving sharable, reusable knowledge bases. *Principles of Knowledge Representation and Reasoning: second internationnal conference.*

Koestler, A. and Smythies, J. (1969). *Beyond reductionism.* Hutchinson.

Prévost, G. (2005). *Modélisation d'écosystème multiniveaux par des approches mixtes*. PhD thesis, Université du Havre.

APPLICATION OF HOMOTOPY PERTURBATION METHOD FOR ECOSYSTEMS MODELLING

Zaid Odibat[1] and Cyrille Bertelle[2]

[1] Prince Abdullah Bin Ghazi Faculty of Science and IT
Al-Balqa'Applied University
Salt, Jordan
email: odibat@bau.edu.jo

[2] LITIS, University of Le Havre
25 rue Ph. Lebon, BP 540
76058 Le Havre cedex, France
email: cyrille.bertelle@univ-lehavre.fr

Keywords: Homotopy perturbation method; ecosystems; Adomian decomposition method; variational iteration method.

ABSTRACT

The HPM method can be considered as one of the new methods belonging to the general classification of perturbation methods. These methods deals with exact solvers for linear differential equations and approximative solvers for non linear equations. In this paper, we focus our attention on the generation of the decomposition steps to build a solver using the HPM method. We present how this method can be used in ecosystem modelling. We develop some solvers for prey-predator systems involving 2 or 3 populations.

1 INTRODUCTION

Ecosystems modelling can be approach in many ways. Global methods are based on differential systems and Individual-based methods allow to represent local phenomena. The first methods are efficient to formulate general behavior of the whole system by using global parameters. The seconds deal with a better understanding on which local phenomenon are hidden inside these global parameters but they lead to high consuming computing. Innovative models are today based on hybrid approaches which manage during computing different way to express the ecosystems behavior. When and where some regularities are observed, we can change from individual-based model to differential ones. When automatic processes are developped in that way, we need both to build automatic equation and identify the parameters value but we need also to solve automatically the generated equations. The method proposed here is the Homotopy Perturbation Method (HPM) which is applied to nonlinear ecosystems.

Usually, perturbation methods need some kind of small parameter to be used. In the HPM method, which doesn't require a small parameter in an equation, a homotopy with an imbedding parameter $p \in [0,1]$ is constructed. The method provides analytical approximate solutions for different types of nonlinear ecosystems. The results reveal that the method is very effective and simple for obtaining approximate solutions of nonlinear systems of differential equations.

The HPM, proposed first by He (He 1999; He 2000), for solving differential and integral equations, linear and nonlinear, has been the subject of extensive analytical and numerical studies. The method, which is a coupling of the traditional perturbation method and homotopy in topology, deforms continuously to a simple problem which is easily solved. This method has a significant advantage in that it provides an analytical approximate solution to a wide range of nonlinear problems in applied sciences. The HPM is applied to Volterra's integro-differential equation (El-Shahed 2005), to nonlinear oscillators (He 2004a), bifurcation of nonlinear problems (He 2005a), bifurcation of delay-differential equations (He 2005b), nonlinear wave equations (He 2005c), boundary value problems (He 2006) , quadratic Riccati differential equation of fractional order (Odibat and Momani 2006), and to other fields. This HPM yields a very rapid convergence of the solution series in most cases, usually only a few iterations leading to very accurate solutions.

2 ANALYSIS OF HPM

The HPM which provides an analytical approximate solution is applied to various nonlinear problems (see the references). In this section, we introduce a reliable algorithm to handle in a realistic and efficient way the nonlinear ecosystems. The proposed algorithm will then be used to investigate the system

$$\begin{aligned} Dx_1(t) &= \textstyle\sum_{j=1}^n a_{1j}(t)x_j + f_1(t, x_1, x_2, \dots, x_n), \\ Dx_2(t) &= \textstyle\sum_{j=1}^n a_{2j}(t)x_j + f_2(t, x_1, x_2, \dots, x_n), \\ &\vdots \\ Dx_n(t) &= \textstyle\sum_{j=1}^n a_{nj}(t)x_j + f_n(t, x_1, x_2, \dots, x_n), \end{aligned} \tag{2.1}$$

subject to the initial condition

$$x_1(0) = c_1, \quad x_2(0) = c_2, \ \dots, x_n(0) = c_n, \tag{2.2}$$

where f_i is a nonlinear function for $i = 1, 2, \dots, n$. In view of the homotopy perturbation technique, we can construct, for $i = 1, 2, \dots, n$, the following homotopy

$$Dx_i(t) - \sum_{j=1}^n a_{ij}(t)x_j = pf_i(t, x_1, x_2, \dots, x_n), \tag{2.3}$$

where $p \in [0, 1]$. The homotopy parameter p always changes from zero to unity. In case of $p = 0$, Eq. (2.3) becomes the linearized equation

$$Dx_i(t) = \sum_{j=1}^n a_{ij}(t)x_j, \tag{2.4}$$

and when it is one, Eq. (2.3) turns out to be the original equation given in the system (2.1). The basic assumption is that the solution of the system (2.1) can be written as a power series in p:

$$x_i = x_i^0 + px_i^1 + p^2x_i^2 + p^3x_i^3 + \dots \quad . \tag{2.5}$$

Substituting Eq. (2.5) into Eq. (2.3), and equating the terms with identical powers of p, we can obtain a series of linear equations of the form

$$\begin{aligned} &p^0 : Dx_i^0(t) = \textstyle\sum_{j=1}^n a_{ij}(t)x_j^0,\ x_i^0(0) = c_i, \\ &p^1 : Dx_i^1(t) = \textstyle\sum_{j=1}^n a_{ij}(t)x_j^1 + f_i^1(t, x^0),\ x_i^1(0) = 0, \\ &p^2 : Dx_i^2(t) = \textstyle\sum_{j=1}^n a_{ij}(t)x_j^2 + f_i^2(t, x^0, x^1),\ x_i^2(0) = 0, \\ &p^3 : Dx_i^3(t) = \textstyle\sum_{j=1}^n a_{ij}(t)x_j^3 + f_i^3(t, x^0, x^1, x^2),\ x_i^3(0) = 0, \\ &\quad\vdots \end{aligned}$$

where the functions $f_i^1, f_i^2, f_i^3, \dots$ satisfy the following equation

$$\begin{aligned} &f_i(t, x_1^0 + px_1^1 + p^2x_1^2 + \dots, \dots, x_n^0 + px_n^1 + p^2x_n^2 + \dots) \\ &= f_i^1(t, x_1^0, x_2^0, \dots, x_n^0) + pf_i^2(t, x_1^0, x_2^0, \dots, x_n^0, x_1^1, x_2^1, \dots, x_n^1) \\ &+p^2f_i^3(t, x_1^0, x_2^0, \dots, x_n^0, x_1^1, x_2^1, \dots, x_n^1, x_1^2, x_2^2, \dots, x_n^2) + \dots . \end{aligned}$$

Setting $p = 1$ in the Eq. (2.5) yields the solution of the system (2.1). It obvious that the above linear equations are easy to solve, and the components $x_i^k, k \geq 0$ of the homotopy perturbation solution can be completely determined, thus the series solution is entirely determined.
Finally, we approximate the solution $x_i(t) = \sum_{k=0}^\infty x_i^k(t)$ by the truncated series

$$\phi_i(t) = \sum_{k=0}^{N-1} x_i^k(t). \tag{2.6}$$

It is also useful, for the system (2.1), to construct the homotopy, for $i = 1, 2, \dots, n$,

$$Dx_i(t) - p\sum_{j=1}^n a_{ij}(t)x_j = pf_i(t, x_1, x_2, \dots, x_n), \tag{2.7}$$

where $p \in [0, 1]$. In this case, the term $\sum_{j=1}^n a_{ij}(t)x_j^0$ is combined with the component x_i^1 and the term $\sum_{j=1}^n a_{ij}(t)x_j^1$ is combined with the component x_i^2 and so on. This variation reduces the number of terms in each component and may minimize the size of calculations. Substituting (2.5) into (2.7), we obtain the following series of linear equations

$$\begin{aligned} &p^0 : Dx_i^0(t) = 0,\ x_i^0(0) = c_i, \\ &p^1 : Dx_i^1(t) = \textstyle\sum_{j=1}^n a_{ij}(t)x_j^0 + f_i^1(t, x^0),\ x_i^1(0) = 0, \\ &p^2 : Dx_i^2(t) = \textstyle\sum_{j=1}^n a_{ij}(t)x_j^1 + f_i^2(t, x^0, x^1),\ x_i^2(0) = 0, \\ &p^3 : Dx_i^3(t) = \textstyle\sum_{j=1}^n a_{ij}(t)x_j^2 + f_i^3(t, x^0, x^1, x^2),\ x_i^3(0) = 0. \\ &\quad\vdots \end{aligned}$$

3 NUMERICAL IMPLEMENTATION

To demonstrate the effectiveness of the HPM algorithm discussed above, several examples of nonlinear systems will be studied. In the first example we choose a linear system to show the features of HPM and the convergence of the homotopy perturbation solution.

Example 3.1 *Consider the linear system (Momani and Odibat 2006)*

$$
\begin{aligned}
x'(t) &= y(t),\\
y'(t) &= 2x(t) - y(t),
\end{aligned} \tag{3.1}
$$

subject to the initial conditions

$$x(0) = 1 \quad , \quad y(0) = -1. \tag{3.2}$$

According to the homotopy given in Eq. (2.7), Substituting (2.5) and the initial conditions (3.2) into the homotopy (2.7) and equating the terms with identical powers of p, we obtain the following two sets of linear equations:

$$
\begin{aligned}
p^0 &: \quad Dx^0 = 0, \qquad x^0(0) = 1,\\
p^1 &: \quad Dx^1 = y^0, \qquad x^1(0) = 0,\\
p^2 &: \quad Dx^2 = y^1, \qquad x^2(0) = 0,\\
p^3 &: \quad Dx^3 = y^2, \qquad x^3(0) = 0,\\
&\quad \vdots
\end{aligned}
$$

$$
\begin{aligned}
p^0 &: \quad Dy^0 = 0, \qquad y^0(0) = -1,\\
p^1 &: \quad Dy^1 = 2x^0 - y^0, \qquad y^1(0) = 0,\\
p^2 &: \quad Dy^2 = 2x^1 - y^1, \qquad y^2(0) = 0,\\
p^3 &: \quad Dy^3 = 2x^2 - y^2, \qquad y^3(0) = 0,\\
&\quad \vdots
\end{aligned}
$$

Consequently, solving the above equations, the first few components of the homotopy perturbation solution for the system (3.1) are derived as follows

$$
\begin{aligned}
x^0 &= 1 \quad , \quad y^0 = -1,\\
x^1 &= -t \quad , \quad y^1 = 3t,\\
x^2 &= \tfrac{3}{2}t^2 \quad , \quad y^2 = t - \tfrac{5}{2}t^2,\\
x^3 &= -\tfrac{5}{6}t^3 \quad , \quad y^3 = \tfrac{11}{6}t^3,\\
x^4 &= \tfrac{11}{24}t^4 \quad , \quad y^4 = -\tfrac{21}{24}t^4,\\
x^5 &= -\tfrac{21}{120}t^5 \quad , \quad y^5 = \tfrac{43}{120}t^5,\\
&\vdots
\end{aligned}
$$

and so on, in this manner the rest of components of the homotopy perturbation solution for the system (3.1) can be obtained. The solution in series form is given by

$$
\begin{aligned}
x(t) &= 1 - t + \tfrac{3}{2}t^2 - \tfrac{5}{6}t^3 + \tfrac{11}{24}t^4 - \tfrac{21}{120}t^5 + \dots,\\
&= \tfrac{2}{3}\Big(1 - 2t + \tfrac{(-2t)^2}{2!} + \tfrac{(-2t)^3}{3!} + \tfrac{(-2t)^4}{4!} + \tfrac{(-2t)^5}{5!} + \dots\Big)\\
&+ \tfrac{1}{3}\Big(1 + t + \tfrac{t^2}{2!} + \tfrac{t^3}{3!} + \tfrac{t^4}{4!} + \tfrac{t^5}{5!} + \dots\Big),
\end{aligned} \tag{3.3}
$$

$$
\begin{aligned}
y(t) &= -1 + 3t - \tfrac{5}{2}t^2 + \tfrac{11}{6}t^3 - \tfrac{21}{24}t^4 + \tfrac{43}{120}t^5 + \dots,\\
&= -\tfrac{4}{3}\Big(1 - 2t + \tfrac{(-2t)^2}{2!} + \tfrac{(-2t)^3}{3!} + \tfrac{(-2t)^4}{4!} + \dots\Big)\\
&+ \tfrac{1}{3}\Big(1 + t + \tfrac{t^2}{2!} + \tfrac{t^3}{3!} + \tfrac{t^4}{4!} + \tfrac{t^5}{5!} + \dots\Big),
\end{aligned} \tag{3.4}
$$

which converges to the exact solution

$$
\begin{aligned}
x(t) &= \tfrac{2}{3}e^{-2t} + \tfrac{1}{3}e^{t},\\
y(t) &= -\tfrac{4}{3}e^{-2t} + \tfrac{1}{3}e^{t}.
\end{aligned} \tag{3.5}
$$

Example 3.2 *Consider the predator-prey system (Momani and Odibat 2006)*

$$
\begin{aligned}
Dx(t) &= x(t) - x(t)y(t),\\
Dy(t) &= -y(t) + x(t)y(t),
\end{aligned} \tag{3.6}
$$

subject to the initial conditions

$$x(0) = 1 \quad , \quad y(0) = 0.5. \tag{3.7}$$

According to the homotopy given in Eq. (2.7), Substituting (2.5) and the initial conditions (3.7) into the homotopy (2.7) and equating the terms with identical powers of p, we obtain the following two sets of linear equations:

$$
\begin{aligned}
p^0 &: \quad Dx^0 = 0, \qquad x^0(0) = 1,\\
p^1 &: \quad Dx^1 = x^0 - x^0y^0, \qquad x^1(0) = 0,\\
p^2 &: \quad Dx^2 = x^1 - x^0y^1 - x^1y^0, \qquad x^2(0) = 0,\\
p^3 &: \quad Dx^3 = x^2 - x^0y^2 - x^1y^1 - x^2y^0, \qquad x^3(0) = 0,\\
&\quad \vdots
\end{aligned}
$$

$$
\begin{aligned}
p^0 &: \quad Dy^0 = 0, \qquad y^0(0) = 0.5,\\
p^1 &: \quad Dy^1 = -y^0 + x^0y^0, \qquad y^1(0) = 0,\\
p^2 &: \quad Dy^2 = -y^1 + x^0y^1 + x^1y^0, \qquad y^2(0) = 0,\\
p^3 &: \quad Dy^3 = -y^2 + x^0y^2 + x^1y^1 + x^2y^0, \qquad y^3(0) = 0,\\
&\quad \vdots
\end{aligned}
$$

Consequently, solving the above equations, the first few components of the homotopy perturbation solution for the system (3.6) are derived as follows

$$
\begin{aligned}
&x^0 = 1, \quad , \quad y^0 = \tfrac{1}{2},\\
&x^1 = \tfrac{1}{2}t \quad , \quad y^1 = 0,\\
&x^2 = \tfrac{1}{8}t^2 \quad , \quad y^2 = \tfrac{1}{8}t^2,\\
&x^3 = -\tfrac{1}{48}t^3 \quad , \quad y^3 = \tfrac{1}{48}t^3,\\
&\vdots
\end{aligned}
$$

and so on, in this manner the rest of components of the homotopy perturbation solution can be obtained. The fourth-term approximate solution for the system (3.6) is given by

$$
\begin{aligned}
x(t) &= 1 + \tfrac{1}{2}t + \tfrac{1}{8}t^2 - \tfrac{1}{48}t^3,\\
y(t) &= \tfrac{1}{2} + \tfrac{1}{8}t^2 + \tfrac{1}{48}t^3,
\end{aligned} \tag{3.8}
$$

which is the same solution for the system (3.6) obtained in (Momani and Odibat 2006) using Adomian decomposition method and variational iteration method.

Example 3.3 *Consider the predator-prey system*

$$
\begin{aligned}
Dx(t) &= ax(t) - bx(t)y(t),\\
Dy(t) &= -cy(t) + dx(t)y(t),
\end{aligned} \tag{3.9}
$$

where a, b, c and d are constants, subject to the initial conditions

$$
x(0) = c_1 \quad , \quad y(0) = c_2. \tag{3.10}
$$

This system is a generalization of the system (3.6). Using the homotopy given in Eq. (2.3), the third-term approximate solution for the system (3.9) is given by

$$
\begin{aligned}
x(t) = {} & c_1 \exp(at) + \tfrac{bc_1c_2}{c}\Big(\exp((a-c)t) - \exp(at)\Big)\\
& - \tfrac{bdc_1^2c_2}{a}\Big(\tfrac{\exp((2a-c)t)}{a-c} + \tfrac{\exp((a-c)t)}{c}\Big)\\
& - \tfrac{b^2c_1c_2^2}{c}\Big(- \tfrac{\exp((a-2c)t)}{2c} + \tfrac{\exp((a-c)t)}{c}\Big)\\
& + \Big(\tfrac{bdc_1^2c_2}{a}\Big[\tfrac{1}{a-c} + \tfrac{1}{c}\Big] + \tfrac{b^2c_1c_2^2}{2c^2}\Big)\exp(at),\\
y(t) = {} & c_2 \exp(-ct) + \tfrac{dc_1c_2}{a}\Big(\exp((a-c)t) - \exp(-ct)\Big)\\
& + \tfrac{d^2c_1^2c_2}{a}\Big(\tfrac{\exp((2a-c)t)}{2a} - \tfrac{\exp((a-c)t)}{a}\Big)\\
& + \tfrac{bdc_1c_2^2}{c}\Big(\tfrac{\exp((a-2c)t)}{a-c} - \tfrac{\exp((a-c)t)}{a}\Big)\\
& + \Big(\tfrac{d^2c_1^2c_2}{2a^2} - \tfrac{bdc_1c_2^2}{c}\Big[\tfrac{1}{a-c} - \tfrac{1}{a}\Big]\Big)\exp(-ct).
\end{aligned} \tag{3.11}
$$

Example 3.4 *Consider the predator-prey system*

$$
\begin{aligned}
Dx(t) &= ax(t) - bx(t)y(t) - cx(t)z(t),\\
Dy(t) &= -dy(t) + ex(t)y(t) - fy(t)z(t),\\
Dz(t) &= -gz(t) + hx(t)z(t) + iy(t)z(t),
\end{aligned} \tag{3.12}
$$

where a, b, c, d, e, f, g, h and i are constants, subject to the initial conditions

$$
x(0) = c_1 \quad , \quad y(0) = c_2 \quad , \quad z(0) = c_3. \tag{3.13}
$$

According to the homotopy given in Eq. (2.3), Substituting (2.5) and the initial conditions (3.13) into the homotopy (2.3) and equating the terms with identical powers of p, we obtain the following two sets of linear equations:

$$
\begin{aligned}
p^0 :\quad & Dx^0 = ax^0, \qquad x^0(0) = c_1,\\
p^1 :\quad & Dx^1 = ax^1 - bx^0y^0 - cx^0z^0, \qquad x^1(0) = 0,\\
p^2 :\quad & Dx^2 = ax^2 - b(x^0y^1 + x^1y^0) - c(x^0z^1 + x^1z^0),\\
& \vdots
\end{aligned}
$$

$$
\begin{aligned}
p^0 :\quad & Dy^0 = -dy^0, \qquad y^0(0) = c_2,\\
p^1 :\quad & Dy^1 = -dy^1 + ex^0y^0 - fy^0z^0, \qquad y^1(0) = 0,\\
p^2 :\quad & Dy^2 = -dy^2 + e(x^0y^1 + x^1y^0) - f(y^0z^1 + y^1z^0),\\
& \vdots
\end{aligned}
$$

$$
\begin{aligned}
p^0 :\quad & Dz^0 = -gz^0, \qquad z^0(0) = c_3,\\
p^1 :\quad & Dz^1 = -gz^1 + hx^0z^0 + iy^0z^0, \qquad z^1(0) = 0,\\
p^2 :\quad & Dz^2 = -gz^2 + h(x^0z^1 + x^1z^0) + i(y^0z^1 + y^1z^0),\\
& \vdots
\end{aligned}
$$

Consequently, solving the above equations, the first few components of the homotopy perturbation solution for the system (3.12) are derived as follows

$$
\begin{aligned}
x^0 &= c_1 \exp(at),\\
y^0 &= c_2 \exp(-dt),\\
y^0 &= c_3 \exp(-gt),\\
x^1 &= \frac{bc_1c_2}{d}\exp((a-d)t) + \frac{cc_1c_3}{g}\exp((a-g)t)\\
&- \Big(\frac{bc_1c_2}{d} + \frac{cc_1c_3}{g}\Big)\exp(at),\\
y^1 &= \frac{ec_1c_2}{a}\exp((a-d)t) + \frac{fc_2c_3}{g}\exp(-(d+g)t)\\
&- \Big(\frac{ec_1c_2}{a} + \frac{fc_cc_3}{g}\Big)\exp(-dt),\\
z^1 &= \frac{hc_1c_3}{a}\exp((a-g)t) - \frac{ic_2c_3}{d}\exp(-(d+g)t)\\
&- \Big(\frac{hc_1c_3}{a} - \frac{ic_2c_3}{d}\Big)\exp(-gt).\\
&\vdots
\end{aligned}
$$

Example 3.5 *Consider the system (Aziz-Alaoui 2006)*

$$
\begin{aligned}
Dx(t) &= a_0x(t) - b_0x^2(t) - \tfrac{\upsilon_0 x(t)y(t)}{d_0 + x(t)},\\
Dy(t) &= -a_1y(t) + \tfrac{\upsilon_1 x(t)y(t)}{d_1 + x(t)} - \tfrac{\upsilon_2 y(t)z(t)}{d_2 + y(t)},\\
Dz(t) &= a_2z(t) - \tfrac{\upsilon_3 z^2(t)}{d_3 + y(t)},
\end{aligned} \tag{3.14}
$$

where a_0, b_0, υ_0, d_0, a_1, υ_1, d_1, υ_2, d_2, a_2, υ_3 and d_3 are model parameters assuming only positive values, subject to the initial conditions

$$x(0) = c_1 \qquad , \qquad y(0) = c_2 \qquad , \qquad z(0) = c_3. \quad (3.15)$$

Multiplying the first equation by the factor $d_0 + x(t)$, the second equation by the factor $(d_1 + x(t))(d_2 + y(t))$ and the third equation by the factor $d_3 + y(t)$. According to the homotopy given in Eq. (2.7), the first few components of the homotopy perturbation solution for the system (3.14) are derived as follows:

$$\begin{aligned}
x^0 &= c_1, \\
y^0 &= c_2, \\
y^0 &= c_3, \\
\\
x^1 &= \Big(a_0 c_1 - b_0 c_1^2 - \frac{\upsilon_0 c_1 c_2}{d_0 + c_1}\Big)t, \\
y^1 &= \Big(-a_1 c_2 + \frac{\upsilon_1 c_1 c_2}{d_1 + c_1} - \frac{\upsilon_2 c_2 c_3}{d_2 + c_2}\Big)t, \\
z^1 &= \Big(a_2 c_3 - \frac{\upsilon_3 c_3^2}{d_3 + c_2}\Big)t. \\
&\vdots
\end{aligned}$$

4 CONCLUSION

In this work, the HPM has been successfully applied to construct approximate solutions for nonlinear systems of differential equations. The method were used in a direct way to study nonlinear ecosystems.

There are some points to make here. First, the HPM doesn't require a small parameter in an equation and the perturbation equation can be easily constructed by a homotopy in topology. Second, the HPM provides the solution in terms of convergent series with easily computable components. Third, the results show that the homotopy perturbation solution in example 1 converges to the exact solution and the approximate solution in example 2 is the same approximate solution obtained using Adomian decomposition method and variational iteration method. Fourth, it is clear and remarkable, from example 3, 4 and 5, that the HPM is effective and simple to solve nonlinear systems, specifically predator-prey systems with 2 or 3 populations. It can be easily generalized to any finite populations number.

References

Abbasbandy S. 2006. "Homotopy perturbation method for quadratic Riccati differential equation and comparison with Adomian's decomposition method" *Appl. Math. Comp.*, 172, 485-490.

Abbasbandy S. 2006. "Numerical solutions of the integral equations: Homotopy perturbation method and Adomian's decomposition method" *Appl. Math. Comp.*, 173, 493-500.

Aziz-Alaoui M.A. . "Complex emergent properties and choas (de-) synchronization". in M.A. Aziz-Alaoui and C. Bertelle (eds), "Emergent Properties in Natural and Artificial Dynamical Systems", Springer, 2006.

El-Shahed M. 2005. "Application of He's homotopy perturbation method to Volterra's integro-differential equation" *Int. J. NonLin. Sci. Mumer. Simulat.,* 6(2) , 163-168.

He J. 1999. "Homotopy perturbation technique" *Comput. Meth. Appl. Mech. Eng.,* 178, 257-262.

He J. 2000. "A coupling method of homotopy technique and perturbation technique for nonlinear problems" *Int. J. Non-Linear Mech.,* 35(1), 37-43.

He J. 2003. "Homtopy perturbation method: a new nonlinear analytic technique" *Appl. Math. Comp.* 135, 73-79.

He J. 2004. "The homtopy perturbation method for nonlinear oscillators with discontinuities" *Appl. Math. Comp.* 151, 287-292.

He J. 2004. "Comparsion of homtopy perturbation method and homotopy analysis method" *Appl. Math. Comp.* 156, 527-539.

He J. 2004. "Asymptotology by homtopy perturbation method" *Appl. Math. Comp.* 156, 591-596.

He J. 2005. "Homotopy perturbation method for bifurcation of nonlinear problems" *Int. J. NonLin. Sci. Mumer. Simulat.,* 6(2), 207-208.

He J. 2005. "Periodic solutions and bifurcations of delay-differential equations" *physics Lettess A,* 374(4-6), 228-230.

He J. 2005. "Application of homotopy perturbation method to nonlinear wave equations" *Chaos, Solitons & Fractals,* 26(3), 695-700.

He J. 2005. "Limit cycle and bifuraction of nonlinear problems" *Chaos, Solitons & Fractals,* 26(3), 827-833.

He J. 2006. "Homotopy perturbation method for solving boundary value problems" *physics Lettess A,* 350(1-2), 87-88.

He J. 2006. "Some asymptotic methods for strongly nonlinear equations" *Int. J. Modern Physics B* 20(10), 1141-1199.

Momani S. and Odibat Z. "Numerical approch to differential equations of fractional order" *J. Comput. Appl. Math.*, in press.

Odibat Z. and Momani S. " Modified homotopy perturbation method: application to quadratic Riccati differential equation of fractional order" *Chaos, Solitons & Fractals*, in press.

Siddiqui A., Mahmood R. and Ghori Q. 2006. "Thin film flow of a third grade fluid on moving a belt by He's homotopy perturbation method" *Int. J. NonLin. Sci. Mumer. Simulat.,* 7(1), 7-14.

Siddiqui A., Ahmed M. and Ghori Q. 2006. "Couette and poiseuille flows for non-Newtonian fluids" *Int. J. NonLin. Sci. Mumer. Simulat.,* 7(1), 15-26.

SIMULATION AND PRODUCTION SYSTEMS

COMPLEX SYSTEMS DYNAMICS IN AN ECONOMIC MODEL WITH MEAN FIELD INTERACTIONS

Gianfranco Giulioni
Department of Quantitative Methods and Economic Theory
University “G. d’Annunzio” of Chieti-Pescara
Viale Pindaro 42, 65127 Pescara, Italy
e-mail: g.giulioni@unich.it

KEYWORDS

economy as a complex system, macroeconomic dynamics, complex dynamics, attractors.

ABSTRACT

In this paper the simulated dynamics of a simple agent based economic system are analyzed. These dynamics are of the complex systems type in the sense that the degree of self-organization changes with time. Indeed the attractor of the macroscopic dynamics changes with time from a cyclic situation to a stable situation and then back again to a cyclic one without changes in the parameters.

INTRODUCTION

One of the most important results of neoclassical Economics, the General Equilibrium Theory, relies on the existence of a coordination mechanism introduced using the elegant device of the walrasian auctioneer. This is probably a provoking sentence, but it opens an important debate in the economic profession: is the presence of coordination mechanisms a good approximation of the economic reality? A positive answer to this question would prevent studies in economics to enter roads already opened for other disciplines like those of the self-organization phenomena and complexity theory. Fortunately, in recent years, a small but growing number of economists became convinced that the economy is a complex system (see Anderson et al. (1988); Arthur et al. (1997); Blume and Durlauf (2005) for example) and therefore started to travel these roads.
Before preparing for the trip proposed in this paper (of course traveling the road we are talking about) some preliminary comments and definitions are useful. In our view, a complex system is composed of a high number of different elements that are interacting in some way, but not through a coordination device. Complex systems are interesting because under certain conditions they give rise to “unusual” dynamics. In particular it is possible that while one of the parameters changes smoothly, the behavior of some endogenous macroscopic variable changes in an unexpectedly organized way or, putting it another way, structures form in an unstructured environment. When such structures emerge without a coordination device researchers generally say that the system “self-organizes”. Sometimes, the expression self-organization is used to denote the emergence of structures in the phase space of dynamical systems. Indeed systems composed of a low number of difference or differential equations can display more structured attractors on the phase space when a parameter is gradually moved. This is not the way the expression self-organization is used in this paper. Dynamical systems are intractable when their dimension increases and consequently they cannot be classified as complex systems (recall that according to our definition a complex systems has a very high dimension). Talking about dynamical systems a confusion may arise because a dynamical system (that, from our point of view, is not complex) can display chaotic dynamics that are usually referred to as “complex dynamics”. Thus, despite the similarity of the expressions, in what follows, “complex systems dynamics” has a different meaning from “complex dynamics”. In particular the latter are a subset of the former at least as long as one identifies chaotic dynamics with the complex dynamics. More interestingly complex systems may exhibit dynamics never detected in dynamical systems. In cellular automata systems, for instance, the existence of such a type of dynamics was found by Wolfram (1986), Langton (1986), and Packard (1988). The last one coined the expression “the edge of chaos” to identify them. We will refer to this type of dynamics as “complex systems dynamics”.

The aim of this paper is to show how the economic system can give rise to “complex systems dynamics.” The model presented below belongs to a set born out of a paper by Greenwald and Stiglitz (1993) (GS hereafter). The intent of these works is to show how the financial conditions of firms is a determinant of the aggregate production of countries, that is, of the Gross Domestic Product (GDP). It is well known that GDP has cyclical dynamics (indeed the explanation of this phenomenon is one of the main topics of macroeconomic theory) and, from a dynamical systems point of view, this calls for the presence of a limit cycle in some relevant variable. GS obtain a difference equation for the financial condition of firms (represented by the equity base) that, under certain parameterization, gives rise to limit cycles and to chaotic dynamics. From our point of view, GS’s work has the serious inconveniece that the millions heterogeneous firms populating the economy are replaced by one of them that is supposed to be representative. This way to proceed is questionable because, among other drawbacks (see Kirman (1992) for example), it limits the analysis to the use of dynamical systems tools that, as maintained above, shuts out complex systems dynamics. The same criticism applies to a paper by Delli Gatti et al. (2000). Building on GS they go a step further recognizing the importance of heterogeneity, but they take it into account introducing a difference equation for the variance of the financial position ending up with the analysis of a two dimensional dynamical system.

In more recent times, GS's type of model have been analyzed using agent based simulation techniques that is, according to us, a more convenient way to deal with complex systems. There are no equations for the macroscopic variables, but only equations governing the individuals' behavior. The values of the macroscopic variables are recovered by simply summing or averaging the individual' ones. Consequently it may happen that the dynamics at the macroscopic level are completely different from those at the individual level identifying genuine emergent phenomena. Delli Gatti et al. (2005) for example show how one can recover particular statistical distributions (basically they are fat tailed distributions like power laws or Weibull) out of the individual data or from the aggregate time series obtained from simulations, and that the same distributions characterize real data. The important observation is that according to a number of scientists the presence of these distributions is common in complex systems dynamics (see Bak (1997) for example).

In what follows we build a model using some "ingredients" from the above cited papers. We then report some simulation results showing how the model produces peculiar dynamics that could be defined as "complex systems dynamics".

THE MODEL

The economy is populated by a large number of firms. As in the GS's type of model we concentrate our attention on the firm. Consumers and others economic agents are supposed to passively accommodate firms' decisions. In these supply side models the production function has a very important role (a large part of the macroeconomic theory is of the supply side type, think for example of the exogenous and endogenous growth and of the real business cycle theories). In the present model, the production function is linear:

$$Y_{it} = \nu_{it} K_{it}$$

where Y_{it} is the production, K_{it} the capital and ν_{it} its productivity.

We dedicate the remainder of this section to the two determinants of the production: ν and K. As mentioned before, we are interested in the emergent properties of the aggregate production dynamics $Y_t = \sum_i Y_{it}$ that we'll recover using a bottom-up approach, that is, through agent based simulations.

Some preliminary notions on the firms variables will be useful and are given here. The balance sheet of a firm is $K_{it} = D_{it} + A_{it}$. where D_{it} is debt and A_{it} the equity base. The fraction $\frac{A_{it}}{K_{it}} = a_{it}$ is the equity ratio that is a signal of the financial soundness of the firm. The dynamics of the balance sheet variables are strictly relative to the economic result of the firm (π_{it}). In these preliminary notions, we restrict ourselves to note that this variable directly affects the dynamics of the equity base in the following way: $A_{it} = A_{it-1} + (1 - \eta_{it})\pi_{it}$ where η_{it} is the fraction of the economic result that does not affect the equity base (more detailed explanation below). The important aspect is that the economic result can be negative and this decreases the equity base. As a consequence, the equity base of a firm could become negative and, if this happens, the firm must leave the market. Another exit mechanism will be considered and we'll come back to this issue below, but what is important to note here is that the presence of exit mechanisms calls for the existence of an entry process. These considerations serve to highlight that an important aspect of this kind of model is the firms turnover (Delli Gatti et al., 2003). However, in this paper we avoid such complication adopting the one-to-one replacement assumption (each exiting firm is replaced by a new one).

Now we can look at the model description starting from the economic result of the firm. In the following steps we use the economic result to determine the dynamics of the two variables we are interested in: the capital (K_{it}) and the productivity (ν_{it}).

Economic result

The economic result (π_{it}) is given by revenues (R_{it}) minus costs (C_{it}):

$$\pi_{it} = R_{it} - C_{it} \tag{1}$$

All the variables are in real terms so that prices will never appear in our equations.

Revenues. Firms sell all their product, but their real revenue from sales may be different from the production due to unforeseen external events (in GS for example this is due to an unknown selling price). We formulate this as follows

$$R_{it} = \nu_{it} K_{it} + u_{it} K_{it} \tag{2}$$

where u is a random variable with mean equal to 0 and finite variance.

Costs. Costs are of two types: production costs (C^L) and adjustment costs (C^K)

$$C_{it} = C^L_{it} + C^K_{it} \tag{3}$$

Production costs. Production costs are due to labor. We use the simplifying assumption that firms need one worker for each unit of capital (Leontief type production function) so that $L_{it} = K_{it}$. Labor costs are

$$C^L_{it} = w_{it} L_{it} = w_{it} K_{it} \tag{4}$$

where w_{it} is the wage and L_{it} the number of employed workers.
Adjustment costs. The adjustment costs must be sustained to adapt the stock of capital (Mussa, 1977). We use here the formulation adopted in Delli Gatti et al. (2000)

$$C_{it}^{K} = \frac{\gamma}{2}\frac{(K_{it} - K_{it-1})^2}{\bar{K}_{t-1}} \tag{5}$$

where $\bar{K}$ is the average capital of the economy. This introduces a first mean field interaction in the model.

The economic result. using equations (1)-(5) the economic result is

$$\pi_{it} = \nu_{it}K_{it} - w_{it}K_{it} - \frac{\gamma}{2}\frac{(K_{it} - K_{it-1})^2}{\bar{K}_{t-1}} + u_{it}K_{it}$$

note that because of the Leontievian assumption, ν_{it} is also the labor productivity. It is natural to think that the wage is related to the (latest known) average labor productivity. From this base we use the following assumption: $w_{it} = \bar{\nu}_{t-1}$ that introduces a second mean field interaction being $\bar{\nu}_{t-1}$ the average productivity of the period before. Under this assumption we can write

$$\pi_{it} = (\nu_{it} - \bar{\nu}_{t-1})K_{it} - \frac{\gamma}{2}\frac{(K_{it} - K_{it-1})^2}{\bar{K}_{t-1}} + u_{it}K_{it} \tag{6}$$

In order to simplify, we avoid discussing the effects of changing the capital level on the economic result. A discussion of these aspects would reveal that the investment involves the movement of the debt stock and affect minimally the economic result. This effect does not modify the behavior of the system and can be eliminated under a further simplifying assumption.

The evolution of Capital

To choose the optimal level of capital the firm maximizes the economic result function, but with two changes. First of all, at the time of the choice, firms don't know the realization of the random variable so that it is replaced with the average value. This allows us to omit the term $u_{it}K_{it}$ being the mean of u equal to zero. Secondly we assume that firms don't know also the average capital and replace it with their own level of capital. Consequently, the objective function to maximize is

$$\pi_{it} = (\nu_{it} - \bar{\nu}_{t-1})K_{it} - \frac{\gamma}{2}\frac{(K_{it} - K_{it-1})^2}{K_{it-1}}$$

Maximizing with respect to K_{it} we have the first element we need, that is, the dynamics of the capital

$$K_{it} = \frac{\nu_{it} - \bar{\nu}_{t-1}}{\gamma}K_{it-1} + K_{it-1} \tag{7}$$

The dynamics of the productivity

The second element we need is the dynamics of the productivity ν_{it}. This variable moves if the firm funds Research and Development activities.

Investment in Research and Development activities. At the end of the period the economic result is realized. It can be positive (profit) or negative (loss).
When a profit is realized the firm has to decide how to use it. We assume that it can be used other than to increase the equity base, to finance Research and Development (R&D) activities. In particular R&D investments are assumed to be

$$R\&D_{it} = \pi_{it}\eta_{it}$$

where η_{it} is the share of profit dedicated to R&D. We assume that this share is an increasing function of the financial soundness of the firm represented by the equity ratio as follows:

$$\eta_{it} = \begin{cases} a_{it-1} & \text{if } \pi_{it} > 0 \\ 0 & \text{if } \pi_{it} \leq 0 \end{cases}$$

From these considerations we can also recover the dynamics of the equity base:

$$A_{it} = A_{it-1} + (1 - \eta_{it})\pi_{it} \tag{8}$$

The dynamics of the productivity. The outcome of the R&D investment is stochastic and the probability of success increases with the amount of funds dedicated to these activities. We formulate this probability as

$$pr = \frac{1}{1 + e^{-b(R\&D_{it} - c)}}$$

where b and c are parameters.
If a firm obtains a success from its R&D activities, its productivity increases by a constant amount β, so that the dynamic of the productivity is

$$\begin{cases} \nu_{it+1} = \nu_{it} + \beta & \text{with probability } pr \\ \nu_{it+1} = \nu_{it} & \text{with probability } 1 - pr \end{cases} \tag{9}$$

SIMULATIONS

We simulate the model using object oriented programming languages. In a first implementation the objective-C version of the SWARM library is used. The validity of the results is checked coding a second time the same model using the RePast java library. We run a large number of experiments to check how the model reacts to changes in the initial conditions, the size of the system (that is the number of firms) and the parameters. Among them, the parameter γ has a very important role. For high values of this parameter the system displays a limit cycle, while cycles disappear for low values. In between, there is a non negligible region where the system gives rise to complex systems dynamics.

We describe here in details one of these experiments where we set $\gamma = 1.5$. The comments below serve also to better explain how the model works. At the beginning the code creates 100000 identical firms giving them the following initial conditions: $K_{i0} = 100$, $A_{i0} = 30$, $\nu_{i0} = 0.1$. The parameters b, c and β are set to 3, 2 and 0.01, respectively.
The algorithm goes through the following steps:

```
1 reset values to firms that meet the exit conditions
2 update the capital
3 update equity ratio
4 update the profit using the random variable
5 update investments in R&D
6 update productivity
7 update equity base
8 collect data
```

Because some steps are technical, we discuss the flow of events in a logical order, this implies that the order reported above will not be respected on some occasions. First of all firms decide their new capital level (step 2) using equation (7): firms with a productivity higher than the average increase capital while the others reduce it. Firms employ the new stock of capital in the production and realize a production equal to $\nu_{it}K_{it}$. Once the production is realized it is sold on the market. The average revenues from sales is equal to production, but some firms realize a higher revenue and some others a lower one due to contingent situations represented by the random variable u that is supposed to be uniform with bounds -0.1 and 0.1. Now, with revenues in their hands the entrepreneurs have to pay their costs: wages and adjustment costs. Here two situations are possible. In the first one, the revenue from sales is higher than costs and consequently a profit is realized. In the second, the revenue from sales is lower than costs and the firm suffers a loss. This is the content of equation (6) implemented in step 4. In step 5 firms with a profit spend a share equal to their equity ratio (that is calculated in step 3) of profit in R&D, while firms that suffer a loss make no expenditures in R&D. After this computation we know for each firm how much they spend in R&D. This allows us to update the productivity using equation (9) in step 6. Moreover, knowing R&D expenditures allows us to determine how the economic result affects the equity base. We do this coding equation (8) in step 7. Finally we record data (step 8) and a new iteration is about to start. At the beginning of the new iteration (step 1), we check the value of the equity base computed in step 7. If it is negative the variables of the firm are initialized with the following values $K_{it} = K_{i0} = 100$, $A_{it} = A_{i0} = 30$, $\nu_{it} = \bar{\nu}_{t-1}$. This can happen to firms that suffer a loss in step 4 of the previous iteration. The fact that they suffer a loss means that they are not able to cover costs with the revenues from sales. At this point they must resort to their internal funds represented by the equity base. In some cases even the equity base is not sufficient to provide the additional needed funds and the firm must exit the market. In addition to this exit mechanism we add also a threshold to the size of the firm, that is, firms with a low level of capital ($K_{it-1} < 20$) but with a positive equity base are also replaced. However, this second exit mechanism is present to catch exceptions and does not affect the simulation results. Finally, note that resetting the variables of the firms when they meet these conditions is the same as assumuming a one-to-one replacement situation and the number of firms is constant to 100000 during the whole simulation. The results are showed in the following graphs and commented on in next section.

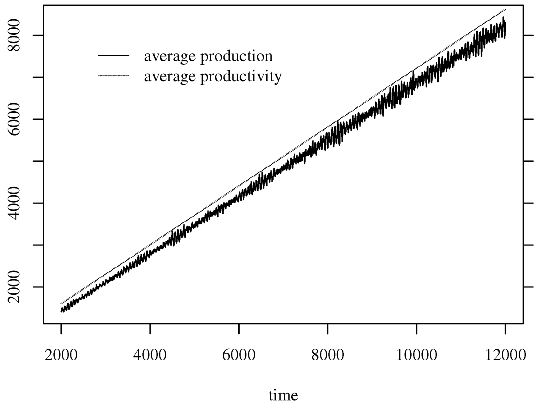


Figure 1: Average production (left axis and black line) and average productivity (right axis and gray line). 2000 time steps have been discarded to eliminate the transient state

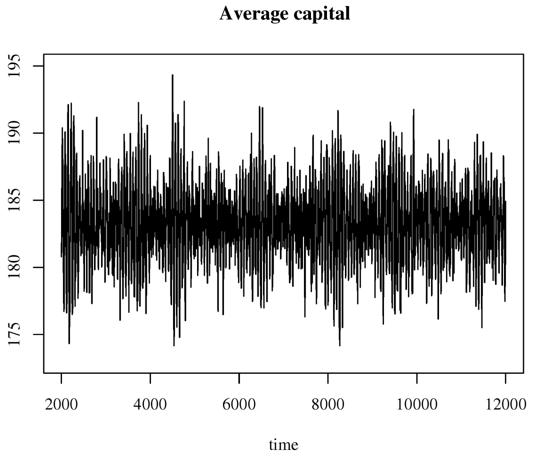


Figure 2: Average Capital. 2000 time steps have been discarded to eliminate the transient state

DISCUSSION

Although the reported graphs could contain interesting features from the economic point of view, the focus of the discussion will be mainly on the type of dynamics a system like this can generate.

Figures 1 and 2 show the dynamics of the variables involved in the production function (production, capital and productivity) for 10000 simulation time steps. We don't show the initial 2000 time steps to avoid the transient state (it occurs in decrising oscillations). In the graphs the average values are reported. Regarding production and capital, one might be interested in the aggregate values. They can be obtained by multiplying those reported in the graphs for the number of firms in the economy that in our case is fixed and equal to 100000. Consequently, the qualitative behaviors of the aggregate production and capital are exactly the same as those reported in the graphs. First of all, from figure 1 it is evident that the increasing trend in production is due to the increasing productivity. Secondly, it is also evident that production is much more volatile than productivity. Having the production function in mind it is straightforward that the volatility of production depends on that of capital; figure 2 confirms this deduction. It is also easy to see that the volatility is not constant, but changes with time in an irregular way. This pushes us toward a more accurate investigation. Figures 1 and 2 report too much data to get an insight into the nature of the volatility by visual inspection.

Figures 3, 4 and 5 shed some light on the phenomenon. In these graphs three contiguous sub-periods with different volatilities are shown.

Figure 3 shows that, in the time span 7750-8750, we are not dealing with stochastic volatility but with a more structured behavior: limit cycles. This is surprising because it is the result of an agent based model with idiosyncratic shocks. As discussed above, obtaining a limit cycle in a dynamical system in not hard, but dynamical systems contemplate the presence of a very low number of equations. In the present model the cyclical behavior is obtained averaging a large number of stochastic equations (one for each firm). From a probability theory point of view, what is reported in the figure is an average of a large number of identical stochastic processes. Figure 3 suggests that the law of large numbers, according to which one expects a very smooth behavior of the average value, does not hold at list in the reported periods. Furthermore, we cannot maintain that this is a feature of the individual behavior preserved at the aggregate level. The uncorrelated idiosyncratic shock present in the model differentiates firms' decisions and, from this point of view, the law of large numbers should apply. A cyclical behavior of the average requires that the various components of the system act in a strong correlated way that, in the absence of a representative agent, could be possible if a coordination mechanism were contemplated. But here we have no coordination device, here each entrepreneur decides alone using its private information (capital and productivity) and the average level of the productivity. Our final conjecture is that in an agent based model the presence of a replacement process and that of mean field interactions (Aoki, 1996) can give rise to a considerable degree of self-organization.

A second observation comes from comparing Figures 3, 4,

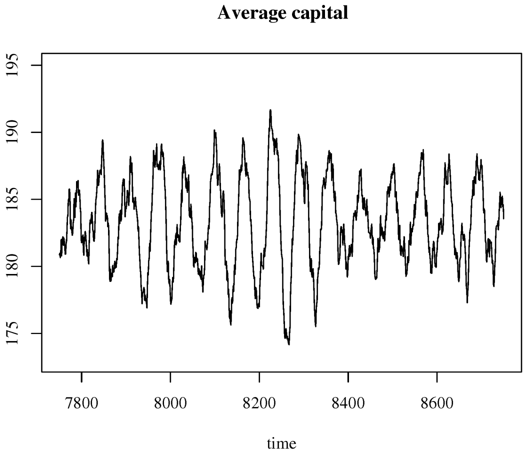


Figure 3: Average Capital from time 7750 to 8750

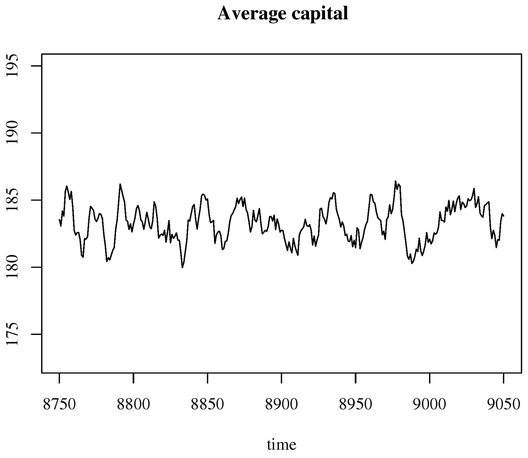


Figure 4: Average Capital from time 8751 to 9050

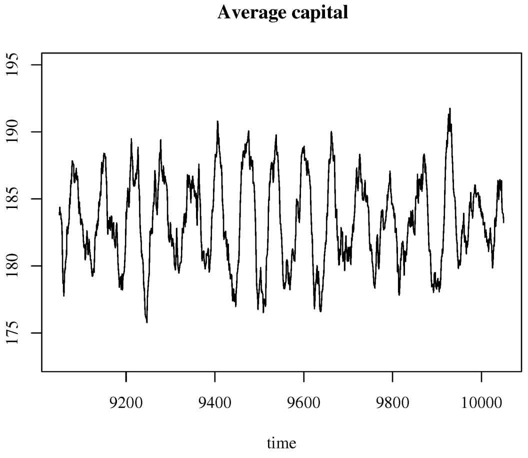


Figure 5: Average Capital from time 9051 to 10050

and 5: the system seems to be able to change attractor in time. The dynamic presented in Figure 4 is quite different from the ones visible in Figures 3 and 5 although they were obtained with the same parameters. In Figure 4 the law of large numbers seems to have a stronger effect than in the other two graphs, that is, the degree of self-organization changes with time. At the actual state of the investigation this phenomenon seems to be a deep emergent property of the system. Indeed one candidate for the explanation could be the average productivity (because it is not fluctuating around a constant value), but looking at its smooth behavior, it seems hard to give it the responsibility to change the system behavior from a limit cycle to (something similar to) an equilibrium and then back to a limit cycle.

The changes in the attractor are also showed in Figure 6 where average values for very short time spans (they are subperiods of Figure 3 and 4) are showed in the equity ratio-capital phase space. It is evident how the economic dynamics can commute between simple (as in the time span 8850-8920) to more structured (as in the time span 8220-8290) attractors. This Figure is also interesting from the economic point of view. Indeed, as discussed in the introduction, GS's type models prove that there is a relationship between the aggregate production and the financial soundness of the economy. Indeed the Figure shows that this relationship exists and is strong in some time spans. Furthermore, looking at the black line in Figure 6, it could be maintained that production and financial fragility move as described by Minsky (1982) in his financial fragility theory of macroeconomic fluctuations. On the other hand, this behavior is not always so strong to be detected as the gray line of the Figure shows.

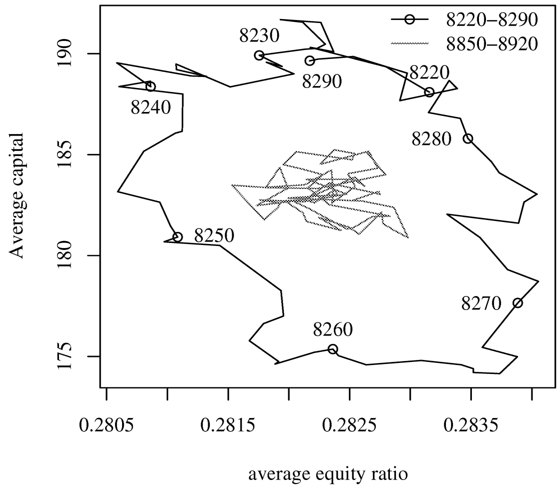


Figure 6: Different attractors in two different time spans

REFERENCES

Anderson, P. W.; K. J. Arrow, and D. Pines, editors. *The Economy as an Evolving Complex System.* New York, Addison-Wesley, 1988.

Aoki, M. *New Aprroaches to Macroeconomic Modeling.* Cambridge University Press, Cambridge, 1996.

Arthur, B. Z.; S. N. Durlauf, and D. W. Lane, editors. *The Economy as an Evolving Complex System II.* Addison-Wesley, 1997.

Bak, P. *How Nature Works. The science of Self-Organized Criticality.* Oxford University Press, Oxford, 1997.

Blume, L. E. and S. N. Durlauf, editors. *The Economy As an Evolving Complex System, III : Current Perspectives and Future Directions.* Addison-Wesley, 2005.

Delli Gatti, D.; C. Diguilmi, E. Gaffeo, M. Gallegati, G. Giulioni and A. Palestrini. A new approach to business fluctuations: heterogeneous interacting agents, scaling laws and financial fragility. *Journal of Economic Behavior and Organization*, 56:489–512, 2005.

Delli Gatti, D.; M. Gallegati, G. Giulioni, and A. Palestrini. Financial Fragility, Patterns of Firms' Entry and Exit and Aggregate Dynamics. *Journal of Economic Behavior and Organization*, 51:79–97, 2003.

Delli Gatti, D.; M. Gallegati, and A. Palestrini. Agent's Heterogeneity, Aggregation and Economic Fluctuation. In D. Delli Gatti, M. Gallegati, and A. P. Kirman, editors, *Interaction and Market Structure*, Berlin,Springer, 2000.

Greenwald, B. C. and J. E. Stiglitz. Financial Market imperfections and Business Cycles. *Quarterly Journal of Economics*, 108:77–114, 1993.

Kirman, A. P. Whom or What Does The Representative Individual Represent. *Journal of Economic Perspective*, 6:117–36, 1992.

Langton, C. Studying Artificial Life with Cellular Automata. *Phisica*, 22D:120–149, 1986.

Minsky, H. P. The financial instability hypothesis: Capitalist processes and the behavior of the economy. In C. P. Kindleberger and J. P. Laffargue, editors, *Financial Crises: Theory, History and Policy.* Cambridge University Press, Cambridge, 1982.

Mussa, M. L. External and Internal Adjustment Costs and the Theory of Aggregate and Firm Investment. *Economica*, 44:163–178, 1977.

Packard, N. Adaptation Toward the Edge of Chaos. Technical report, Center for Complex Systems Research, University of Illinois, 1988.

Wolfram, S. *Theory and Applications of Cellular Automata.* World Scientific, Singapore, 1986.

BIOGRAPHY

GIANFRANCO GIULIONI gained a PhD in Economics at Università Politecnica delle Marche (Italy) in 2001. He has been assistant professor in Economics at Università "G. d'Annunzio" di Chieti-Pescara (Italy) since 2005 where he teaches Public Economics and Economics of Monetary and Financial Markets. His main research interest is applying tools from Complex Systems Theory to explain Macroeconomic phenomena.

COMPLEXITY OF TRAFFIC INTERACTIONS: IMPROVING BEHAVIOURAL INTELLIGENCE IN DRIVING SIMULATION SCENARIOS

Abs Dumbuya[1], Anna Booth[1], Nick Reed[1], Andrew Kirkham[1], Toby Philpott[1], John Zhao[2] and Robert Wood[2]
[1]TRL, Crowthorne House, Nine Mile Ride, Wokingham, Berkshire, RG40 3GA, UK,
[2]Loughborough University, Loughborough, Leicestershire, LE11 3TU, UK
Email: adumbuya@trl.co.uk

KEYWORDS
AI-supported simulation, neural network, simulators, Behavioural science, Psychology

ABSTRACT

This paper introduces modelling concepts and techniques for improving behavioural intelligence and realism in driving simulation scenarios. Neural Driver Agents were developed to learn and successfully replicate human lane changing behaviour based on data collected from the TRL car simulator.

INTRODUCTION

The design and development of realistic scenarios for driving simulators (see discussion on emerging issues relating to the realism of visual databases and scenarios, Parkes, 2005; Allen, 2003 and 2004) could greatly enhance the realism with which simulator trials can be created, since the autonomous vehicles would be capable of responding in a realistic manner both to the behavioural responses of the participant and to any pre-programmed autonomous vehicle behaviour (e.g. a vehicle programmed to disobey a red traffic light). This in turn would improve participants' immersion in simulator scenarios, increasing the likelihood that they will drive in a realistic and representative manner with the consequence that greater confidence can be placed in resulting analyses. This paper describes a research project at TRL which extended previous work (development of a Synthetic Driving SIMulation, SD-SIM framework) conducted at Loughborough University (Dumbuya and Wood 2003; Dumbuya et al. 2002). The paper demonstrates the development and application of a novel technique for improving and verifying the realism of a Neural Driver Agent (NDA) modeling technique which is able to show behavioural intelligence. The technique used an artificial neural network to control the behaviour of a vehicle in a simple lane changing task.

Artificial neural network (ANN) models use a mathematical model for information processing with a functional architecture that resembles the neuron structure of the human brain (for an introduction to the subject see Gurney, 1997). These models are capable of learning from training examples and demonstrating learned behaviour in unseen situations. Abdennour and Al-Ghamdi (2006) applied ANN for the estimation of vehicle headways using data collected from different freeways in Riyadh, Saudi Arabia. Using the collected data they were able to model and train an ANN capable of estimating headways as a function of time (time series prediction) and headways as a general probability density function. Lin et al. (2005) considered some sophisticated artificial neural network architectures, to model human driver behaviour in vehicle handling compared with a Driver-Vehicle-Environment (DVE) system.

The aim of the project reported in this paper was to find whether the vehicle control of the neural network based approach was an improvement over a more traditional rule-based algorithm.

TRL CAR SIMULATOR

TRL's driving simulator uses a real Honda Civic family hatchback that has had its engine and major mechanical parts replaced by an electric motion system that drives rams attached to the axles underneath each wheel. These impart limited motion in three axes (heave, pitch, and roll) and provide the driver with an impression of the acceleration forces and vibrations that would be experienced when driving a real vehicle. All control interfaces have a realistic feel and the manual gearbox can be used in the normal manner. Surrounding the simulator vehicle are large display screens onto which are projected the images that represent the external environment to the driver. The level of environmental detail includes photo-realistic images of buildings, vehicles, signing, and markings, with terrain accurate to the camber and texture of the road surface. The driving environment is projected onto three forward screens to give the driver a 210º horizontal forward field of view whilst a rear screen provides a 60º rearward field of view, thus enabling normal use of all mirrors. Realistic engine, road, and traffic sounds complete the virtual setting. Scenario specification for the behaviour of all autonomous traffic vehicles included in simulated scenarios is determined by applying specific programming commands via SCANeR (Champion et al, 1999).

DEVELOPMENT OF A NEURAL DRIVER AGENT

A multilayer NDA has been designed and implemented. Figure 1 shows a typical architecture of the NDA. The concept of the NDA is based on Artificial Intelligence (AI) techniques, e.g. ANN, which, at a minimum level, aims to model the system such that the system can exhibit human-like properties, for example, planning, learning, knowledge, reasoning and decision making. The NDA is based on supervised learning paradigm, with a *backpropagation* training algorithm. The inputs are propagated through the two hidden layers and output layer. The error (or mismatch) between the output and the pre-specified desired output is minimised by using a *gradient descent* rule (which essentially attempts to avoid the local minimum in training the network, by moving the weights in a direction opposite to the direction of the gradient). This allows the errors to be signalled backwards from output to input nodes until the error approaches zero. In other words, the network learns by adapting interconnecting weights.

To develop the NDA, input and output parameters to the network are defined below. Note that these parameters have been carefully selected to allow straightforward interfacing with the simulator module. A generalised expression can be derived for the multilayer NDA architecture, using gradient decent to approximate the desired output values for new direction and speed. A full derivation may be found, for example, in Gurney (1997). The Neural Driver Agent mathematical expressions derived were implemented in the commercial off-the-shelf NeuroSolutions software package from NeuroDimension (www.nd.com)

- Current speed, v
- Current direction d
- Distance from vehicle dhw
- Current Lane l
- Preferred speed v_p
- New direction d_n
- New speed v_n

RESULTS

The first part of the project was to generate the training data that would be fed to the neural network in order for the network to 'learn' how to change lanes to overtake another vehicle as illustrated in Figure 2.

Eight participants were recruited to complete a short drive on a simulated two lane motorway in the driving simulator. In this drive they were required to accelerate to a constant target speed, remaining in lane 1 of the motorway until they came across an autonomous vehicle travelling at a constant speed also in lane 1. Behaving as they would on a real UK motorway, the participant had to overtake this vehicle by moving to lane 2 and then return back to lane 1. This completed the simulator task. The data from each completed simulator run was used to train the network in how to control the driven vehicle. At each time step, the network evaluated a number of inputs, including current speed, current lane, and distance headway to the vehicle ahead, to generate outputs of desired speed and desired direction of the driven vehicle, which could be used to calculate the new position and direction of the driven vehicle. The network outputs were compared to the actual changes in speed and direction observed from the real drivers. With sufficient training the network was able to cause the driven vehicle to follow a realistic path at a suitable speed around the lead vehicle.

Comparison of Neural Drivers and Real Drivers

Figures 3 show the changes in direction produced by the Neural Driver Agent (NDA) and real drivers when performing an overtaking manoeuvre at a speed of 70mph. The direction scale is in degrees such that 0 is straight ahead, a negative value is steering to the right and a positive value is steering to the left. The graphs show how the drivers steer to the right to move into the middle lane, then steer to the left to move back into the inside lane. The graphs show how differently real drivers perform an overtaking manoeuvre and how individual drivers also produce different behaviour at different speeds. Despite the differences in driving behaviour produced by the real drivers, the graphs show that the NDA has learnt the changes in direction required to perform an overtaking manoeuvre.

Figure 4 show the real drivers and Neural Driver Agent (NDA) accelerating to and trying to maintain a speed of 70mph. The graph also demonstrates the differences in the behaviours of the real drivers. When trying to achieve a speed of 70mph the NDA accelerates too much but then decelerates to maintain a speed just less than 70mph. However, overall the NDA produces a smooth acceleration and can maintain a constant speed.

Assessing Behavioural Realism – Results from Driving Simulator Study

The second part of the project was to demonstrate that the neural network model was more realistic in its control of the driven vehicle than the rule-based model (from previous Loughborough University research) performing the same task. To achieve this, twelve participants were recruited to observe how each micro-simulation model controlled the driven vehicle and to rate the realism with which they thought the vehicle was being controlled. Participants each sat in the driver's seat of the simulator vehicle and were effectively 'driven' by the micro-simulation models through the simulated scenario on which the neural network model had been trained. Furthermore, participants also observed a pre-recording of how a human driver had completed the same manoeuvre. Participants were asked to rate how realistic they felt each model was on a ten-point scale from 1 to 10, where a rating of 1 indicated that they felt that the model was very unrealistic and a rating of 10 indicated that they felt that the model was very realistic. In rating the realism of the three computer models of driver behaviour, Figure 5 shows that on average, the human drive was the most realistic (average score of 7.83) and SD-SIM was the least realistic (average score of 2.92). It is important to note the realism of the NDA in replicating the human drive (average score of 7.42). The results of the study demonstrated that participants thought that the neural network model was significantly more realistic in its control of the driven vehicle than the traditional rule-based model. Paired samples t-tests showed that the realism scores for SD-SIM differed significantly from those given for the NDA ($t(11) = 7.24$; $p < 0.001$) and from those given for the Human ($t(11) = 11.3$; $p < 0.001$), whilst the comparison of the NDA and the Human realism scores did not reach significance ($t(11) = 0.767$; $p = 0.46$).

To further explore the realism of the three models, participants were asked to rate how likely it was that each of the three drives presented to them was actually completed by a human driver. Again, a ten-point scale from

1 to 10 was used, where a rating of 1 indicated that they felt that it was very unlikely that the drive was completed by a human driver and a rating of 10 indicated that they felt that it was very likely that the drive was completed by a human driver. The aim was to see if participants could correctly distinguish the human drive from the SD-SIM and NDA-based drives. Figure 6 illustrates the participants' performance in classifying the models. Furthermore, participants were unable to discriminate between the human and the neural network in their control of the driven vehicle.

DISCUSSION OF RESULTS AND CONCLUSIONS

In general the comparison of human and neural driver agent results is good. However, there is a noticeable difference in SD-SIM and the other two models. This is contributed to by three factors: (1) fine tuning of driver characteristics in SD-SIM is currently a demanding task (2) the current vehicle model in SD-SIM lacks some of the detailed inertia and frictional effects found in the steering and suspension of real vehicles so, (3) creating a driver character in SD-SIM to match real vehicle behaviour implicitly involves some compensation for this. However, it should also be emphasised that only a small component of SD-SIM's Intelligent Virtual Driver (IVD) has been assessed. For example, the vision model which provides distance and speed estimation capabilities in lane changing behaviour was not considered in this project. Furthermore it is important to note that SD-SIM provides a framework which allows future enhancements and additions of components to improve realism. This was part of the reason for proposing the neural driver to replace some of the rule-based approaches adopted in SD-SIM. Important outcomes of the research included:

1. The study explored the use of Artificial Intelligence (AI) theories and techniques such as Artificial Neural Network (ANN) to develop a new Neural Driver Agent (NDA) model. The model was trained using captured behavioural data from participants in the simulator and demonstrated increased 'intelligence' of traffic interaction.

2. To assess behavioural intelligence and realism in driving simulation scenarios, participants rated model realism on a scale from 1-10. The results showed that human driver was the most realistic (average score of 7.83). The NDA performed well by learning and replicating human drive (average score of 7.42).

3. To further explore the realism of the models, participants were asked to rate how likely it was that each of the drives presented to them was completed by a human driver. Participants were unable to discriminate between the human and the neural network in their control of the driven vehicle.

In terms of the implications of the results in developing behavioural intelligence and realism in driving scenarios results, NDAs could potentially be developed to participate fully in driving scenarios and allow them to respond realistically in both situations to which they have been trained and novel situations. By adjusting the structure and/or connection weights of the neural network, it may also be possible to create simulated aggressive drivers, tired drivers, alcohol-impaired drivers, learner drivers, and so on. This will help to represent the range of behaviours displayed by real drivers in driving scenarios. In fact new work is underway to validate the NDA with over 50,000 individual vehicle data collected from the UK motorway.

ACKNOWLEDGEMENT

The authors would like to thank Lena Weaver, Simulator Trial Manager for her help in conducting the simulator trials and TRL technical reviewer of the paper, Prof. Andrew Parkes for his useful comments on the paper. We are also grateful to all the participants who took part in the trials. We are particularly grateful to TRL Science and Engineering Directorate (Prof. Rod Kimber), now TRL Academy (Neil Paulley) for funding this work and to Bob Wood and John Zhao of Loughborough University for their collaboration on the project. The views expressed in this paper belong to the authors and are not necessarily those of TRL or Loughborough University.

REFERENCES

Abdennour, A. and Al-Ghamdi, A.S., (2006). *Artificial neural networks applied to the estimation of vehicle headways in freeway sections,* **TEC,** February 2006, pp. 56-59.

Allen, R.W., Park, G., Rosenthal, T.J., and Aponso, B.M. (2004). *A process for developing scenarios for driving simulations*, **IMAGE 2004** Conference, Arizona, Paper No. 632.

Allen, R.W., Rosenthal, T.J., and Park, G. (2003). *Scenarios produced by procedural methods for driving research, assessment and training applications*, **Driving Simulation Conference, North America (DSC-NA)**, Michigan, Paper No. 621.

Champion, A., Mandiau, R., Kolski, C., Heidet, A. and Kemeny A. (1999). *Traffic generation with the SCANeR II simulator: towards a multi-agent architecture*, **Driving Simulation Conference, DSC'99** Paris, France, pp. 311-324

Dumbuya, A.D., Wood, R.L., Gordon, T.J., and Thomas, P., (2002). *An agent-based traffic simulation framework to model intelligent virtual driver behaviour.* **Proceedings, Driving Simulation Conference (DSC'02)**, Paris, France, pp.363-373, 11-13, September 2002.

Dumbuya, A. D. and Wood, R.L. (2003). *Visual perception modelling for intelligent virtual driver agents in synthetic driving simulation*, **Journal of Experimental and Theoretical Artificial Intelligence (JETAI)**, vol.15, no.1, pp.73-102

Gurney, K., (1997), *An introduction to Neural Networks*, UCL Press

Lin Y., Tang P., Zhang, W.J. and Yu, Q. (2005). *Artificial neural network modelling of driver handling behaviour in a driver-vehicle-environment system.* **International Journal of Vehicle Design**, Vol. 37, No.1 pp. 24 - 45

Michon, J.A. (1985). *A Critical view of driver behaviour models: What do we know, what should we do?* in L. Evans and R. Schwing, (eds), **Human Behaviour and Traffic Safety**, (London: Plenum), pp 516-520.

Parkes, A. M. (2005). *Improved realism and improved utility of driving simulators: are they mutually exclusive?* **HUMANIST Workshop**, Conference on Application of New Technologies to Driver Training. CDV, Brno, Czech Republic, January 2005.

FIGURES

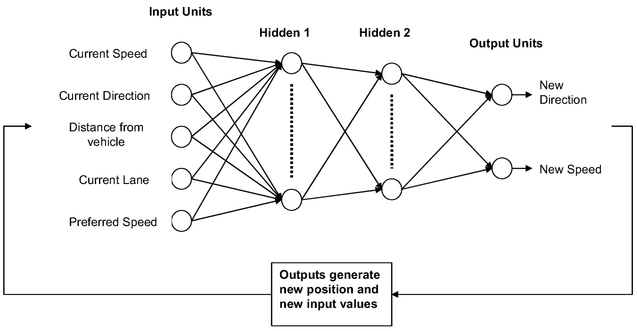


Figure 1: Neural Driver Agent (NDA) architecture

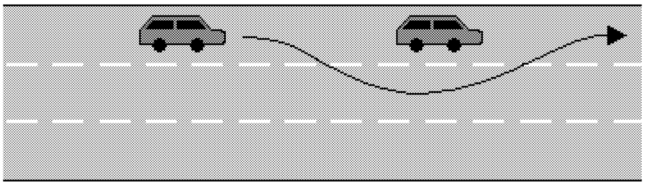


Figure 2: Scenario set-up to generate training data for NDA

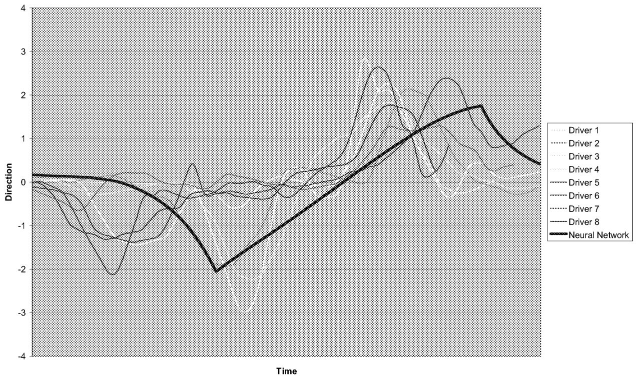


Figure 3: A graph to show the drivers' change of direction when overtaking at a speed of 70mph

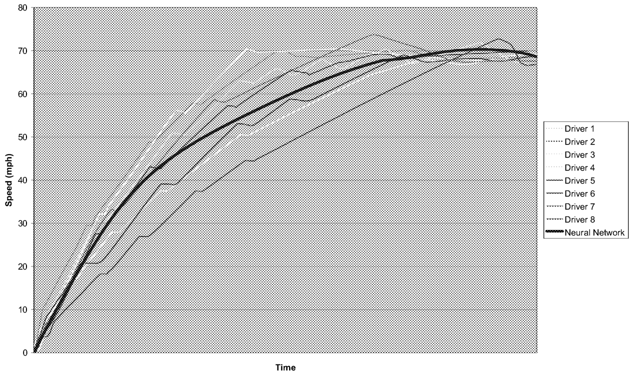


Figure 4: A graph to show the drivers' speed when trying to maintain a speed of 70mph

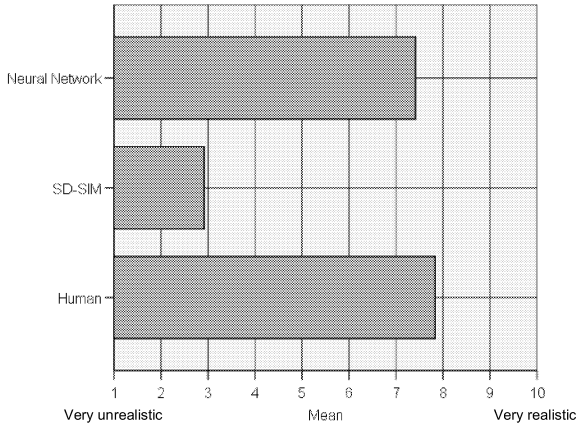


Figure 5: A graph to show the mean realism ratings of each of the models presented to participants

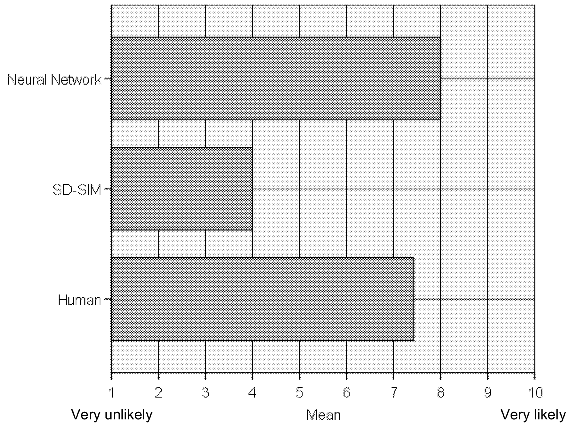


Figure 6: A graph to show participants' mean ratings of how likely it was that each drive was actually completed by a human driver

An Integrative Simulation Model for Project Management in Chemical Process Engineering

Bernhard Kausch
Nicole Schneider
Morten Grandt
Christopher Schlick
Chair and Institute of Industrial Engineering and Ergonomics
RWTH Aachen University
D 52062, Aachen,
Germany
E-mail: {B.Kausch, N.Schneider, M.Grandt, C.Schlick}@iaw.rwth-aachen.de

KEYWORDS
Decision Support, Forecasting, Discrete Simulation, Simulation Interfaces, Event oriented, Work Organization, Workflow Simulation

ABSTRACT

The planning of development projects significantly influences the costs created by the projects as well as the success of the development projects. The presented approach shows a method for the modeling and simulation of development projects in process engineering based on Petri net simulation. The simulation of an example process displays the connections between different influencing parameters such as team configuration, the availability of needed tools, the variance in processing times, and the qualification of the persons involved. It could hereby be determined with which parameter combination and with which amount of employed staff the shortest development time can be attained. In the outlook several additional parameters are named that are prepared and will be added to in the further research project in order to make further detailed analyses possible.

INTRODUCTION

Only 13% of work in projects in Germany are actually value-adding, out of which a total loss of value of approximately 150 billion Euros results (Gröger 2006). Reasons for these deficits are bad decisions in the selection of projects, yet also the insufficient defining of goals. While these problems affect the project environment in the business there is also another area that affects the project structure. This area covers the development and continued use of findings and information in projects. This, along with the accurate implementation of employee competence and availability, must be improved through workflow planning. These problems are well-known in process engineering as well. The project correlations in the process development were analyzed in the Collaborative Research Center (CRC) 476. Besides these theoretical results eight out of 12 project managers from the field of process engineering that were surveyed said that a lack of coordination and poor information flow were the main causes for sub-optimal project efficiency. Support tools were developed that improve the cooperation of the various development areas and that are meant to reduce the interface-related losses. A simulation system identifies the necessary correlations and information flows between the organizational units involved based on a semi-formal project model. The simulation clarifies the connection between the assigned resources and persons, thereby making the identification of the project duration possible through defined input of resources or vice versa. As a result, the project planner has the possibility to analyze different workflow management structures and then plan the input of resources or the resource-relevant project structure accordingly.

DEVELOPMENT PROJECT IN PROCESS ENGINEERING

An example project was recorded that reproduces typical workflows of process engineering to take into consideration the current requirements within process engineering alongside the procedure models described in literature. The development of the synthetic material Polyamide 6 (PA6), which is usually used in the manufacturing of textiles, yet also as friction and heat resistant construction material, represents the characteristics of process engineering development processes.

Process development usually begins with literature research that serves the purpose of information collection, and which is frequently repeated in the development project. Based on the collected information, yet also based on the experience of the developers involved, the decision for the batch or the continuous operation is made. This decision influences the additional procedure-dependent development steps. In the case of our example, the development of the PA6 process was performed in cooperation with chemical engineering companies, the developments of the reaction, separation and extrusion follow. These developments result individually, yet depend heavily on each other, founding the basis of the complexity within the development projects of chemical engineering. The development of the facility area necessary for the various steps is based on the representation of the mathematical, chemical and physical correlations. Consequently, the main task in here is the creation and analysis as well as the improvement of these models. To conclude, the final decision regarding the plant concept is made based on the simulation results of previous work steps.

Definition of the Simulation Approach

A workflow simulation model of development projects in the chemical industry was developed at our institute in recent years. One way to differentiate simulation models is by the level of detail found in human modeling. VDI-Guideline 3633 distinguishes between person-integrated models (person as reactive action model) and person-oriented models (display of various additional traits possessed by person) (VDI-Guideline 1993, Zülch 2004). Furthermore, simulation models of product development processes can, similar to VDI-Guideline 3633, be differentiated by two forms of model logic:

1. In the case of actor-oriented simulation models, system dynamics are produced by actors (modeled persons or organizational units) based on specific task (Steidel 1994, Christiansen 1993, Cohen 1992, Jin and Levitt 1996, Levitt et al. 1999, Licht et al. 2004).

2. In process-oriented simulation models system dynamics are produced by activities through the usage of resources (persons, tools) (Browning et al. 2000, Cho et al. 2001, Cho et al 2005, Gil et al. 2001, Raupach 1999).

According to this terminology, the workflow simulation model in process engineering that will be presented here can be characterized as a person-oriented and process-oriented approach.

EXISTING SIMULATION APPROACHES

In the field of process and product development processes only a few adequate simulation techniques are well established.

The so-called Virtual Design Team (VDT) is an actor-oriented model especially designed for the simulation of product development projects which was created by Levitt's research group at Stanford University. Early versions of the VDT were already able to model actors and tasks, as well as the information flow between these two instances (Christiansen 1993, Cohen 1992). Subsequent versions then also took into consideration the different goals of actors, the construction of exceptions, and in addition, exception handling (Jin et al. 1996, Levitt et al. 1994, Levitt 1999). A process engineering context is not considered in this model, and participative creation of the simulation model or optimization of workflow management will not be supported through the methodology.

Independent of Levitt's group, Steidel managed to develop a further actor-oriented simulation model for product development processes (Steidel 1994). This model also ignores particularities of process engineering. Likewise, participative creation of the simulation model or optimization of workflow management will also not be supported through the methodology.

Raupach formulated a process-oriented approach for the simulation of product development processes so that consistency can be observed in various construction solutions. The product structure is accounted for in great detail through this approach (Raupach 1999). This fact makes it hard to apply in contents with inherent variability e.g. the process engineering context, participative process creation, and optimization of workflow management. Those points are not addressed. Interdependencies between project success criteria and factors influenceable by technical planning will not be examined in this approach.

Eppinger's research group at the Massachusetts Institute of Technology developed numerous process-oriented simulation models (Browning and Eppinger 2000, Cho and Eppinger 2001, Cho and Eppinger 2005).

Browning's simulation model assumes that an unlimited supply of resources (in this case, employees) exists, meaning the simulation results of this model are limited in their representation of reality. Cho's simulation model does take note of the limitation of resources available in a product development project, yet a corresponding processing of multiple activities is also not possible in this case. An organizational connection to process engineering is non-existent, and participative process creation or an optimization of workflow management is not intended. Interdependencies between project success criteria and factors influenceable by technical planning will hardly be considered.

A process-oriented model for the simulation of a factory-planning project was developed by a research group headed by Tommelein at the University of California at Berkeley (Gil et al. 2001). This model observes the effects of altered requirements on the planning process and the project length of construction projects. Particularly, examination of so-called postponement-strategies occurs, in which the start of a succeeding operation is purposely delayed in order to increase the quality of the work results of the preceding-operation. Similarly, the simulation model assumes an unlimited supply of resources. However, in a process engineering context, participative process creation or an optimization of workflow management is not dealt with. Interdependencies between the technical planning of influenceable factors and project success criteria are not sufficiently taken into consideration in this model.

The person-centered simulation model of Licht (Licht et. al. 2004) offers an, according to our requirements, more suitable approach to analyzing development processes of products and processes. The model includes many different process specific aspects of the process, such as type and complexity of products, characteristics of the employees, tools, organizational structure, etc. Due to the person-oriented approach, the model also serves as a realistic method for employee management by providing employees' behavior. The negative consequence, however, is that the model is very complex and therefore difficult to apply.

INTEGRATIVE SIMULATION MODEL

The simulation model presented here offers a suitable technique for project planners in order to compare several alternative ways of project organization at an early stage, with respect to the number of persons, tools, time and other resources involved.

With its close connection to the easy to understand and semi-formal modeling language C3 (Killich et al. 1999), designed at our institute, the model enables a transparent and very concise, understandable and well applicable representation, making it easy for the user to understand and

to work with the simulation model. The goal of this simulation model is to combine the advantages of C3 (Eggersmann et al. 2001, Schneider et al. 2006) and the advantages of the simulation, that is to say, the possibility of planning, analyzing and rearranging the development process based on mathematical constraints. In addition, the model used offers the chance to optimize the development process with respect to the development duration as well as in consideration of resources and the development costs.
The entire simulation model is based on the following five partial models:
1.) the task network, 2.) the task, 3.) the employee, 4.) the work tool, and 5.) the information, which will be examined in greater detail in the following.

Task Network Model

The development of a new or modified chemical process usually takes place in team spanning development projects. It is in these projects that the complexity concerning the organizational structure as well as the workflows should be reduced. The model concept of the task network describes the workflow management of the development project. In addition, the individual phases of the development process will be divided into work tasks through the use of a workflow plan (a so-called task network). Predecessor-successor-relationships, i.e., the logical order of execution - for example, due to causal relationships between individual underlying activities - of the tasks will be laid down in the task network. Hereby it is determined, for example, that the literature research precedes the additional analysis. The workflow plan is primarily participatively recorded and displayed through the C3 modeling language. The work tasks of the task network are assigned to organizational units for execution. Apart from the chronological sequence of tasks, the assignment of work equipment for the associated tasks is also displayed in the task network. An overview of the PA6 development process, described earlier, with 79 tasks is schematically displayed in Figure 1.

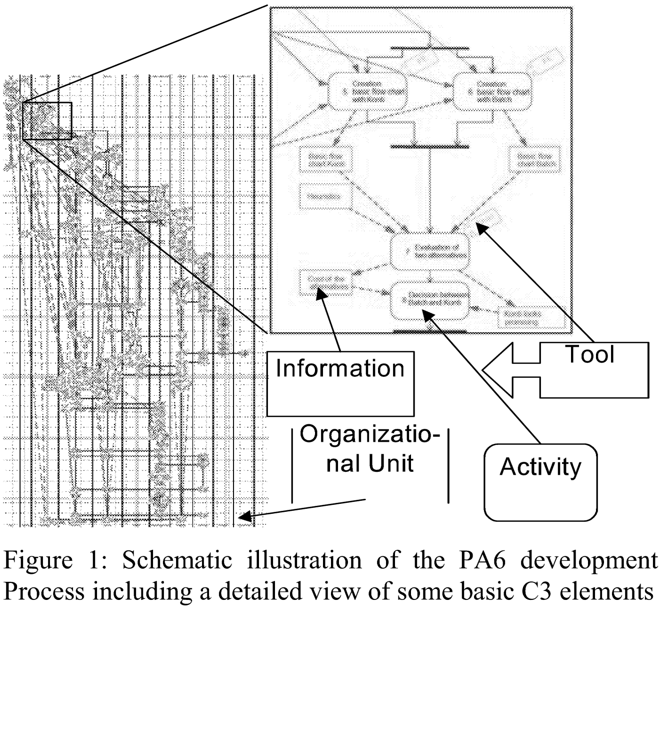


Figure 1: Schematic illustration of the PA6 development Process including a detailed view of some basic C3 elements

Additionally in the extract on the upper right a detailed view into the process is given, where the main elements of C3 are marked and briefly explained. A software environment, especially designed in the research project to support recording and visualization of work processes with the C3 method, supports the recording process as well as the visualization and software based transformation of the working process structure. In Schneider et al. (2003) this software environment for work process modeling is described in more detail.

Task Model

The task network consists of the tasks in the development process that need to be worked on. The processing of each individual task is described in detail in this model concept. Within the tasks there is information about the subject matter needing to be processed, a necessary work tool, a profile of possible persons to do this processing, input and output information of the task as well as the expected duration needed to process the task. For the processing of a task, a qualified person and, if necessary, adequate tools are selected to achieve the goal of only implementing the most qualified employee actually available for the handling of the task. Each person is then also assigned a value that reflects the quality of the person, dependent on the task at hand and the required tools to complete the task. This value is calculated from the weighted sum of the person's assigned characteristics (cp. Model Concept of the Employee).
The weighting and the different attributes are not constant and can be varied depending on the area of application. The most highly qualified person will then take on the task, though it may occasionally be the case that the basic skills needed for a certain task are not possessed by anyone. In such an event, the task cannot be completed until someone suitable for the task becomes available. Only once the adequate labor and essential work tools are available the task can be carried out according to its duration, which depends on the underlying distribution function and the person employed for the task.

Employee Model

According to the person-oriented basic approach of simulation, the definition of the characteristics of employees and thus the participants in this model concept is of particular importance. At the same time, an attempt is made to model the person as realistically as possible. This entails displaying employees' characteristics and abilities that have an influence on the allocation of persons to the various tasks as well as the task processing time and work quality of the different development process tasks. The described attributes of an employee are summarized in the following:

- Productivity of an Employee

Each person is assigned a numerical value that describes the individual productivity, i.e., output. This value improves the quality of the employee in the selection of the most qualified employee for a task, and also has an influence on the processing time of a task.

- Qualification in terms of a particular area of work

The tasks of the development process are arranged into swim lanes in accordance with C3 modeling. These swim lanes describe the areas of work, for example, such as in the PA6 Processes case study in which the work areas of Simulation or Separation were described. The persons possess abilities and skills that qualify them for the processing of tasks in certain areas of work, yet then also make them unsuitable for others.

- Ability to deal with particular work tool

Several tasks require a work tool such as a software tool or a machine for their processing. The persons possess abilities and qualifications that describe how well they can handle certain work tools. This means a person must not only bear the appropriate qualifications to complete the task, but they must also have the ability to carry out the task through use of the necessary work tools.

- Learning aptitude

An employee begins his career with certain basic qualifications, i.e. abilities that were acquired during schooling, or inherent characteristics. During the course of a career, however, a person's abilities can change. Due to routine tasks and new methods and expertise, certain qualifications can actually be improved. Alternatively, abilities not put to use over a greater period of time can also be weakened. This capacity to learn and unlearn is shown in a simulation model through a learning curve that is attributed to each person. A more detailed description of the learning curve will be presented later on.

Personal qualifications and abilities are taken into account in the model concept in terms of recognizing that each person is able to act out a variety of activities. This portfolio of possible activities can be directed at specific job descriptions that are representative of the different organizational units and work means related to the process.

Work Tool Model

The influence of work tools on the completion of tasks in the scope of the execution of activities through an employee is held in the partial model of work tools. The allocation of work tools to tasks results through the work organization of the development project. Simultaneously, the information of which work tools can be used for which task is already retained in the model of the task network. Due to their scarcity, work tools must be reserved prior to their use. Also, a tool can be used by only one employee at a time, though more than one tool can be used for a specific task. The amount of possible work tools cannot be exhaustively declared since the amount of possible tasks in need of completion, detached from individual case examples, cannot be fully indicated. Thus, similar to the task network and the work organization in relation to the development project that is to be simulated, the list of work tools must be created and must be specific. The level of detail is also to be specified individually for each case. This means that it may be enough in a project to simply differentiate between work tools for the creation of technical drawings between drawing board and CAD; in other projects, due to the use of varying computing systems and thereby related file formats, there must be distinction between different computing systems.

Information Model

Information should be viewed in the same light as work tools. Information is already assigned to tasks in the task network and has an influence on the duration of the development project. Information can be grouped into input and output information. Input information describes files or documents that are necessary for the processing of a task. The processing of a task cannot start without this information.

For example, for task seven of the case example (cp. Figure 1), evaluation of two alternatives for the creation of a basic flow chart with Batch or Konti requires information about various heuristics as well as output from the basic flow charts of Batch and Konti. These can either be produced in the form of output information through a different task, such as the Batch or Konti information which is linked to the previous tasks, or be made available outside of the analyzed workflow as in the case of the heuristics.

Through the processing of a task, output information is treated as its result. The results of a task, which may eventually be needed for the processing of later tasks, are described and are made available as input information.

IMPLEMENTATION OF THE SIMULATION MODEL

To show the implementation of the simulation model, the Polyamide 6 process (Eggersmann 2004) was used as an example case of the CRC. The underlying process here, consisting of 79 activities executed by the coordination between eight organizational units (separated by swim lanes in the C3 model), describes the different phases of new development for the manufacturing of PA6.

To maintain the distinctiveness of the C3 language the simulation model was implemented using a person-oriented and process-oriented approach. Also, to formally describe the simulation model, the notation of Timed Stochastic Colored Petri Nets was taken up. The development project was mapped into a directed graph consisting of places, transitions, arcs, and markings. A great advantage of this simulation notation is that a stepwise simulation can easily identify weak points. In this case, Petri Net tokens as representatives for active elements indicate the status of work progress, and indicate it as a result of possible weak points.

The simulation model was implemented using the Petri Net Simulator Renew (Kummer et al. 2004). Renew is a Java-based high-level Petri Net simulator developed at the Department of Informatics at the University of Hamburg. The simulation tool provides a flexible modeling approach based on reference nets as well as a user-friendly design by the use of a graphical presentation. Renew is a computer tool that supports the development and execution of object-oriented Petri Nets, which include net instances, synchronous channels, and seamless Java integration for easy modeling.

The entire Petri Net model according to the description of the Polyamide process is composed of different sub-networks that correspond to partial models that, for instance, represent the universal model. The implementation of this partial model in the form of sub-networks will be examined more closely in the following.

Task Network

The Task Network describes the workflow management of the development project. The predecessor-successor-relationships between individual tasks are defined in the corresponding Petri Net. Certain tasks are released for further processing through appropriate transitions in this network when all necessary predecessor tasks have been completed and the adequate persons as well as resources (work tools, input information) for the processing are available. A section of the task network of the PA6 Process is displayed in Figure 2. Based on the process-oriented approach, the task network builds the link between the partial models. Here are the rough correlations, such as how the development project implements workflow management and the necessary resources for the processing of individual tasks, whereby the exact processing of tasks are represented in the network of the task.

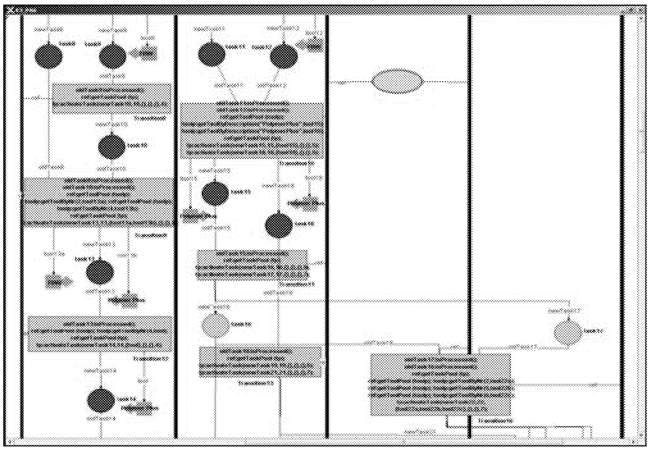


Figure 2: Screenshot of a section of the PA6 Process

Task

The net for the representation of the processing of a task builds the link between the partial model of the work tool and the employee. Here, the person who will process the task is chosen and the necessary resources are reserved.

In doing so, the basic conditions are directed at the person who is qualified for the processing of the task. These requirements are implemented in the respective task and organized according to the area of application, with the most qualified person executing the task. The qualification level (Q_L) is calculated as follows:

$$Q_L = \alpha P + \beta Q_w + \gamma Q_t$$

The weights α, β and γ determine how strong the influence of an attribute is on the quality level of a person. According to the model concept of the person, the attributes productivity P, qualification based on the field of work Q_w and the ability and qualification to handle a work tool Q_t are viewed as influencing variables. Moreover, the duration of a task is determined and thus processed in the network of the task. Effort and duration for the processing of a task depend on the estimated average processing duration as well as the qualification and proficiency level of the specific employee. The choice of work tools used along with the procurement of additional information can also have an effect on the duration and processing of a task. In order to realistically depict the processing time of a task, which can only be approximated, the aid of a probability distribution is employed. A normal distribution with relative variance between 10% and 30% of the mean was established for the first test runs of the simulation model.

The administration of the tasks of the workflow is implemented in the Task Pool. The Task Pool is a help network that, in combination with the Task Net, displays a task on the model concept. The various tasks are initialized and managed in the Task Pool.

Person

The employees involved in the project, inclusive of their characteristics and capabilities, are implemented in the Person Net. The management of employees is organized in an auxiliary net, the so-called Person Pool. Here, the current number of available persons as well as their current status - ”currently in processing” or ”free for the next available task” - is deposited. Before a task can be processed, however, a search occurs in the net for the fitting employee for the processing of the task. (cp. the Net of a Task).

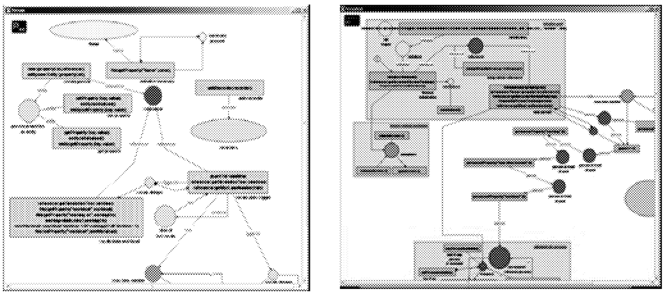


Figure 3: Screenshot of the Renew Person Net and the Person Pool

In Figure 3 a section of a screenshot of the Person Net as well as the Person Pool is mapped. Task-specific abilities of a person are improved, thereby increasing the attributes of that person when a task is processed. This learning ability of the employee is implemented through a learning curve as follows (figure 4):

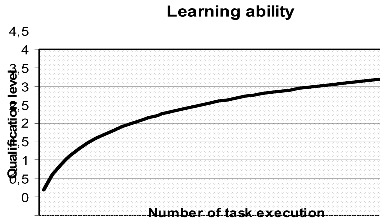


Figure 4: Coherence between individual Qualification and number of Task executions

In the implementation of the learning curve it was not possible to deal with the learning ability of every single employee, though a general function was implemented.

The Tool Net

The work tools available for the work process are administered in the Tool Net and the Tool Pool. In the model presented here, a name and a distinct identifier are sufficient as a characteristic of a work tool. Modeled on the Person Pool, the Tool Pool implements the maintenance of work tools, that is to say, the current number as well as status of available work tools is accounted for. It may sometimes be the case that another person is already using this specific tool, leading to waiting times.

Implementation of Additional Functions

A universal model composed of further help networks exists in addition to the networks that describe partial models. In this universal model functions, such as the initialization of the model or the output of simulation model results, are implemented. These act as links between the various nets.

The input data of the simulation model (the description of the tasks in the development project, including their demands of the employee as well as their necessary resources, the amount and attributes of the employees involved in the project, as well as the work tools available for the project) is organized in tables and can be viewed with the help of the initialization network.

Additional functions, for example, the calculation of the normal distribution of the processing time or the printout of simulation results, are implemented in independent Java classes whose functions are invoked and performed in corresponding parts of the network, more precisely, in the transitions.

RESULTS AND DISCUSSION

Concerning the structural validation of the simulation model, the coordination of the numerous individual parameters among each other should be seen as particularly critical. These parameters produce extremely complex system dynamics through which the investigation and evaluation of the models is in turn made more difficult.

In the first test runs of simulation the number of persons was varied and afterward set to the optimal number. Following this, the number of tools was also varied. The influence of these factors on the simulation time was then examined in order to judge the validity of the simulation model. To do so, the expected durations of the individual tasks were acquired in multiple expert workshops.

As described in the following, these initial test runs showed satisfactory behavior.

Examination of Dependence between Number of Employees and Total Project Duration

The relationship between the total duration of the development project and the number of organizational units working on them - in the present case identical to the persons working on the task - was analyzed in the first simulation runs. Also, it was assumed in the form of the simplest case, that only one person processed a task. For this comparison the amount of persons was varied between one and 11. The variance of the expected duration was still regarded as an independent variable and then changed in three steps, between 10%, 20% and 30% of the mean, so that ten (n=10) runs will be simulated for each of the possible 33 (b=33 out of: 11 differing amounts of people x 3 differing variances) combinations of variables. The corresponding hypothesis states that the duration decreases with each additional employee. Experts forecast that the influence of the number of employees will far exceed the boundary-defined duration variable. Therefore, it was the total duration, forecasted through the simulation model, which was to be analyzed.

Simulation Result

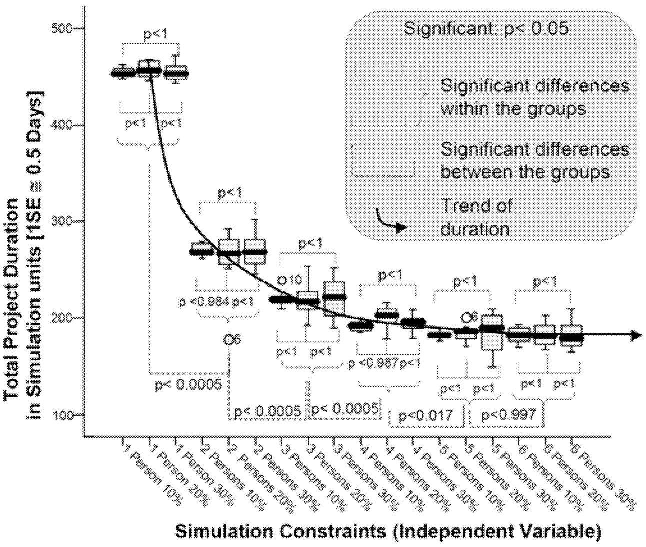


Figure 5: Dependencies between the total Project Duration and the number of employees and the variance of the expected value of activity duration time

The results (see Figure 5) were first examined on the basis of significant differences in duration. Through a one-way analysis of variance (ANOVA, $\alpha = 0.05$) it is shown that there is no significant difference within the groups that have the same number of employees. This confirms the hypothesis that with any of the possible deviations from the expected value of duration time (between 10% and 30%) that are regarded in the simulation, no significant change in total duration time takes place. This is supported independently of the predicted duration and describes a balancing effect on the variance of a large number of activities (a=79). Experience shows, however, that projects usually do encounter delays, which is why the variance in the re-design of the simulation should be replaced by a right-skewed β-distribution.

The simulation further shows that the duration can be reduced by approximately 60% through employment of more than five persons. After employment of more than six persons though, no significant reduction in duration can be measured. This is due to the project structure’s task network in which no more than five tasks can be carried out at the same time, thereby also not being able to be processed by more employees. These circumstances change when more persons can be employed simultaneously for the processing of specific activities. Task sharing within a task is promoted through this, and the resulting implications were examined in further studies. Subsequent reasons for an unwanted short duration through a high number of employed organizational units lie in synchronous communications. These occur in specific intervals, lying between the tasks, and thereby occupy the required persons of the participating organizational units. In doing so, the employees are picked from the task network and "scheduled" for the discussion through the simulation. These employees can process no other tasks during this time. These communication relationships are a particular feature of development processes so that the high significance assigned to them through the simulation corresponds to actual conditions.

Examination of Dependency between Number of Work Tools and Total Duration

According to results of the first examination, the parameter "number of persons" was set to the optimal number of six employees (see Figure 6).

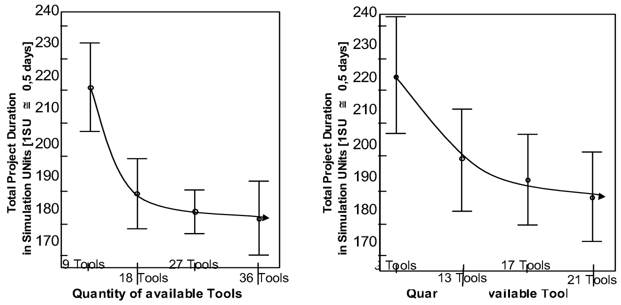


Figure 6: Connections between the simulated duration and the quantity of tools

Due to the non-significant differences concerning the distribution of variance of the expected project duration (see Figure 5) it was set to 30%. Ten simulation runs were conducted for each of the set parameter combinations. In the variation of work tools, however, their total quantity does not play a crucial role; instead, the number of very specific work tools, depending on the structure of the project, does.

Thus, the minimum requirements of the selected example process (in this case, Polyamide 6) are met for processing of various tasks for which the process' nine different work tools are needed, and for which each work tool must be available at least once.

Moreover, several work tools are needed only once or only in a work area with sequentially run tasks; additional work tools of the same type then no longer have a positive influence on the duration of the project.

To substantiate this fact, two different procedures took place. In first runs the quantity of all nine work tools was increased; test runs with nine, 18, 27 and 36 work tools were accomplished. Next, the same was done for the quantity of four selected work tools whose increase in number could also have an influence on the duration of the example process due to its structure (runs with nine, 13, 17 and 21 work tools).

Simulation Results

The results of both of these approaches are displayed in Figure 5. The hypothesis that the simulation time is not based solely on the number of work tools, rather on their properly combined quantity, can clearly be seen in the diagram. The results of the test runs in which all tools were duplicated without any preconceived expectations are displayed on the left side and the other ones on the right.

In the second runs only four of the nine work tools were added in each case since the remaining five work tools would only be needed once in the entire project, or in sequentially occurring tasks of an area.

The naive duplication of the 18 work tools produces the same results as a quantity of 13 work tools of which only the most necessary ones were duplicated. The same can be said for the work tool quantities of 27 and 17, as well as 36 and 21.

When the process of the respective test runs is taken into consideration, it can be noted that after a quantity of 17 work tools, or as the case may be, 27, no further significant improvement in simulation time can be achieved. A slight regressive tendency can be seen when there is an additional increase in quantity. This can be explained through the structure of the project in which apparently no more than three identical work tools are needed simultaneously.

FUTURE RESEARCH

The presented approach to the research project will be further developed in the future in close collaboration with enterprises in the chemical engineering industry. In addition, attributes and correlations that were not contained in the theoretically created example process, and to which, along with iterations and probability rich decisions, the formation of tool and task groups with similar job specifications belong, are added. The current scattering task processing times following the normal distribution must, in support of the phenomenon that tasks tend towards longer processing times, be changed through a right-skewed beta distribution. This and several other parameters are determined and quantified via ergonomic methods. Furthermore, the correlations of individual factors are empirically calculated through the modeling of several example processes.

The planned sensitivity analysis serves the validation of the simulation results and is to make possible a transfer of the realizations to planned work processes.

The long term goal is a round planning support through project combinations capable of simulation in order to increase the validity and the time and resource planning. Thus, improved risk management in daily project planning is also allowed for.

ACKNOWLEDGEMENTS

The research was funded by the German Research Foundation according to the collaborative research center no. 476, Improve.

REFERENCES

Browning, T.R., Eppinger, S.T.: Modelling the Impact of Process Architecture on Cost and Schedule Risk in Product Development:, Massachusetts Institute of Technology, Sloan Scool of Management, Working Paper No. 4050, Cambridge, MA 2000

Cho, S.-H., Eppinger, D.: Product Development Process Modeling Using Advanced Simulation, In: "Proceedings of DETC'01, ASME 2001 Design Engineering Technical Conferences and Computers and Information in Engineering Conference", 9-12 September 2001, Pittsburgh, PA 2001

Cho, S.-H., Eppinger, D.: A Simulation-Based Process Model for Managing Complex Design Projects, In: IEEE Transactions on Engineering Management, 52, 3, S. 316–328, August 2005

Christiansen, T.:Modeling Efficiency and Effectiveness of Coordination. In Engineering Design Teams, PhD thesis, Stanford University, Palo Alto, CA,USA, 1993.

Cohen, G.: The Virtual Design Team: An Object-Oriented Model of Information Sharing in Project Teams, Ph.D. Thesis, Stanford University, Palo Alto, CA, 1992

Eggersmann, M.; Schneider, R., Marquardt, W.: Modeling Work Processes in Chemical Engineering – from recording to supporting, Technical report LPT-2001-31, 2001

M. Eggersmann: Analysis and Support of Work Processes Within Chemical Engineering Design Processes, Published in: Fortschritt-Berichte VDI, Nr. 840, VDI-Verlag, Düsseldorf, 2004

Gil, N., Tommelein, I.D., Kirkendall, R.: Modeling Design Development Processes in Unpredictable Environments, In: Proc. 2001 Winter Simulation Conference. Invited Paper in the Session, Extreme Simulation: Modeling Highly-Complex and Large-Scale Systems. Online im Internet: URL: http://www.informs-sim.org/wsc01papers/067.PDF [Stand 26.01.2006], 2001

Gröger, M.: Wertschöpfungspotenzial Projektmanagement In: REFA-Nachrichten 1/2006, Pp. 4-7, ISSN 0033-6874

Jin, Y., Levitt, R.; The Virtual Design Team: A Computational Model of Project Organizations. Computational and Mathematical Organization Theory 2:3 171-195. 1996

Killich, S.; Luczak, H.; Schlick, C.; Weissenbach, M.; Wiedenmaier, S.; Ziegler, J.: Task modelling for cooperative work. In: Behaviour & Information Technology, Hampshire, 18 5, S. 325-338, 1999

Kummer, O., Wienberg, F., Duvigneau, M., Schumacher, J., Köhler, M., Moldt, D., Rölke, H., Valk, R.: An Extensible Editor and Simulation Engine for Petri Nets: Renew. In: Proceedings of Applications and Theory of Petri Nets 2004: 25th International Conference. 484–493, 2004

Kusiak, A. and H.H. Yang. 1993. "Modeling the Design Process with Petri Nets". In Concurrent Engineering 1993, H. Parsei and W.G. Sullivan (Eds.). Chapman & Hall, London.

Köhler, M.; Mold, D.; Rölke, H.; Spresny, D.: Handlung und Struktur. Modellierung von Akteurmodellen. In *Sozionik – Modellierung soziologischer Theorie*, R.v. Lüde; D. Mold; R. Valk (Eds.). Lit Verlag, Münster, 2003

Levitt, R.E.; G.P. Cohen; J.C. Kuntz; C.I. Nass; T. Christiansen; and Y. Jin.: The Virtual Design Team: Simulating How Organizational Structure and Information Processing Tools Affect Team Performance, In Computational Organization Theory 1994, K.M. Carley and M.J. Prietula (Eds.). Lawrence Erlbaum Assoc., Hillsdale, N.J., 1994

Levitt, R., Thomson, J. Christiansen, T., Kunz, J., Jin, Y. Nass, C.: Simulating Project Work Processes and Organizations:Toward a Micro-Contingency Theory of Organizational Design.Management Science, Informs 45:11; 1479-1495. 1999

Licht, T., Dohmen, L., Schmitz, P., Schmidt, L., Luczak, H.: Person-Centered Simulation of Product Development Process using timed stochastic colored Petri-Nets. In: Proceedings of the European simulation and Modeling Conference, 2004

Raupach, H.-C.: "Simulation von Produktentwicklungs-prozessen." Dissertation, TU Berlin, Berlin, 1999

Schneider, N.; Kausch, B.: Simulationsgestützte Optimierung der Organisationsgestaltung in Entwicklungsprozessen. In: Innovationen für Arbeit und Organisation, Bericht zum 52. Arbeitswissenschaftlichen Kongress vom 20. - 22.3.2006 am Fraunhofer - IAO Stuttgart, Hrsg.: Gesellschaft für Arbeitswissenschaft e.V.. GfA-Press, Dortmund 2006, Pp. 431-436. 2006

Schneider, R.; Gerhards, S.: WOMS - A Work Process Modeling Tool In: Nagl, M., Westfechtel, B. (Hrsg.): "Modelle, Werkzeuge und Infrastrukturen zur Unterstützung von Entwicklungsprozessen", Wiley VCH, Weinheim, 375-376, 2003

Steidel, F.: "Modellierung arbeitsteilig ausgeführter, rechnerunterstützter Konstruktionsarbeit – Möglichkeiten und Grenzen personenzentrierter Simulation." Dissertation, TU Berlin, Berlin. 1994

VDI-Richtlinie VDI 3633: Simulation von Logistik-, Materialfluss und. Produktionssystemen, Dez. 2001

Zülch, G., Jagdev, H., Stock, P., eds.: Integrating Human Aspects in Production Management, Springer, 2004

BERNHARD KAUSCH studied engineering at the Technical University of Munich. His area of specialization was ergonomics and product development. Since August 2002 he is research assistant at the Institute of Industrial Engineering and Ergonomics at RWTH Aachen University.

NICOLE SCHNEIDER studied computer science at the RWTH Aachen University and completed her diploma degree in 2004. Data management and exploration was her area of specialization during her main study period. Since April 2005 she is research assistant at the Institute of Industrial Engineering and Ergonomics at RWTH Aachen University.

MORTEN GRANDT received his M.S. degree (Dipl.-Ing.) in Safety Engineering from Wuppertal University in 1995, and the Ph.D. degree (Dr.-Ing.) also in Safety Engineering from Wuppertal University in 2004. From 1995 he worked as a scientific assistant and later on as provisional head of the Ergonomics and Information Systems department at the Research Institute for Communication, Information Processing and Ergonomics, Wachtberg, Germany. He is now head of the department Human-Machine-Systems at the Institute of Industrial Engineering and Ergonomics at Aachen University of Technology. He is chairman of the Ergonomics Chapter of the German Aerospace Society (DGLR) and member of NATO-RTO task groups.

CHRISTOPHER M. SCHLICK received his M.S. degree (Dipl.-Ing.) in Electrical Engineering from Berlin University of Technology in 1992, Ph.D. degree (Dr.-Ing.) in Mechanical Engineering from Aachen University of Technology in 1999, and the Habilitation degree (Dr.-Ing. habil) also in Mechanical Engineering from RWTH Aachen University of Technology in 2004. He worked for the computer networks industry in 1992 and 1993 as a design engineer. From 1994 to 2000, he joined the Institute of Industrial Engineering and Ergonomics at RWTH Aachen University of Technology. From 2000 to 2004 he was the head of the Department of Human–Machine Systems at the Research Institute for Communication, Information Processing and Ergonomics, Wachtberg, Germany. He is now a full professor of Industrial Engineering and Ergonomics at Aachen University of Technology.

LATE PAPERS

Advanced Discrete HMM Network Structures for Classification and Prediction

Professor Costas Xydeas
Department of Communication Systems
Infolab21
Lancaster University
United Kingdom
E-mail: c.xydeas@lancaster.ac.uk

KEYWORDS
HMM Networks, Pattern Classification and Data prediction, EEG signal processing, Voice over IP.

ABSTRACT

Pattern classification and data prediction are generic, key elements of modern information processing systems whose general function is to analyze received input application specific information, in order to achieve situation awareness and thus provide a decision making capability that can be i) offered to users in an advisory mode or ii) employed directly to control associated and often particularly complex systems and processes. This paper focuses on the HMM probabilistic paradigm of signal classification in general and presents the design methodology of two new Discrete HMM modular network structures that offer flexible input data fusion characteristics, improved classification performance as well as the means for dynamically quantifying the importance of each input data stream. Furthermore, the proposed classification and prediction methods are applied, using computer simulation, to two different application domains; namely: i) in a medical related application for the classification of the pain/ no pain condition of subjects using EEG signals and ii) in a Voice over IP application for the prediction and recovery of missing, due to lost data packets, vocal tract related data decoding parameters..

INTRODUCTION

Pattern classification and data prediction are generic, key elements of modern information processing systems whose general function is to analyze received input application specific information, in order to achieve situation awareness and thus provide a decision making capability that can be i) offered to users in an advisory mode or ii) employed directly to control associated and often particularly complex systems and processes.

In general, situation awareness should be seen as a hierarchical process having several levels in the hierarchy; each level accepts information that emerges at the output of the previous level and effectively forms a more abstract/higher level, application specific understanding of the current situation.

In this sense, a data classification and prediction system design methodology should be modular and amenable to the formation of hierarchical structures, while at the same time it exhibits robust performance characteristics in terms of the "quality"/purity" and "completeness" of the input data.

In addition, and when operating in non stationary, dynamically changing application environments, it should easily offer itself towards integration/combination with an input signal segmentation process that determines stationary segments in a way that maximizes overall system classification/ prediction performance. It is also highly desirable for the system design methodology to allow for a dynamic, changing with time appreciation of the importance or otherwise of the many and often different type/modality of "feature signals" or "data streams" used as input data to the classification/prediction process.

Within this general scenario for the deployment of pattern classification and data prediction, this paper focuses on the HMM probabilistic paradigm of signal classification in general and presents the design methodology of two new Discrete HMM modular network structures that offer flexible input data fusion characteristics, improved classification performance as well as the means for dynamically quantifying the importance of each input data stream.

Furthermore, the proposed classification and prediction methods are applied, using computer simulation, to two different application domains; namely: i) in a medical related application for the classification of the pain/ no pain condition of subjects using EEG signals and ii) in a Voice over IP application for the prediction and recovery of missing, due to lost data packets, vocal tract related data decoding parameters..

HMM BASED CLASSIFIERS

Hidden Markov Models (HMM) is a rigorous probabilistic classification framework that has been successfully applied in several application areas in general (Rabiner 1989) and speech recognition in particular (Bengio 1999). Furthermore, its natural capability of dealing with time varying patterns of arbitrary lengths is attractive from a real applications point of view, due to the expected variability in the time lengths of patterns/signals to be classified.

In its Discrete format, a Discrete HMM (HMM-D) modelling process can be described as a probabilistic network having *N* hidden states $\{S_1,S_2,\ldots,S_N\}$ and *M* possible observations can be generated by each state at a given instant in time .Thus at every time step one of the states, say S_j, is entered based on the state transition probability $\{a_{ij}\}$ which depends on the previous state S_i.

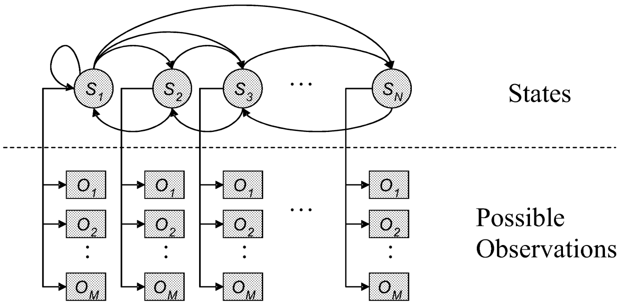


Figure 1. A single HMM structure with *N* Hidden States and *M* possible observations associated per state

After each transition is made, an observation, say the *m*-th observation O_m, is produced from S_j , see figure 1, with corresponding observation probability $\{b_j(O_m)\}$, note that the initial state probabilities are defined as $\{\pi_i\}$. A compact notation $\lambda=\{\{a_{ij}\},\{b_j(O_m)\},\{\pi_i\}\}$ is set to indicate the parameters of the network that effectively models the underlining observations producing process . Therefore given an observation sequence $O=\{o_1,o_2,...,o_T\}$, that is obtained over a period of time *T,* the maximum likelihood probability $P(O|\ \lambda)$ associated with the λ th model can be calculated by tracing in a trellis of hidden states the Viterbi path $Q=\{q_1,\ q_2,\ .....q_T\}$ that is most likely to generate $O=\{o_1,o_2,...,o_T\}$.

Alternatively, the assumption of discrete observations can be replaced by one where observations O_m and observation transition probabilities $\{b_j(O_m)\}$ are represented by a continuous (usually Gaussian) density process

$$f_{jk}(O_t)=\frac{1}{\sqrt{(2\pi)^L|U_{jk}|}}e^{-\frac{1}{2}(O_t-\mu_{jk})'\cdot U_{jk}^{-1}\cdot(O_t-\mu_{jk})} \qquad (1)$$

where $f_{jk}(O_t)$ is a Gaussian density with mean μ_{jk} and a covariance matrix $U_{jk,}$ or a mixture of Gaussian densities (Huang et al 1990).

Note that in many applications, observation vectors are "continuous" signals and HMM-C schemes offer high classification performance. In other applications however, vector-format input observations are not continuous, correlation coefficients of continuous densities are uncorrelated or nearly close to zero, the matrices U_{jk} are not singular (Moon and Sterling 2000) and $f_{jk}(O_t)$ estimates can be inaccurate.

Factorial HMMs (FHMM) are a generalization of the Continuous Density HMMs (HMM-C) modeling approach where hidden states are factored into multiple state variables i.e. a hidden state q_t is now represented and therefore dependent on R state variables (Ghahramani 2001). The FHMM formulation can provide a performance advantage over traditional HMMs (Bourlard and Morgan 1998).

A major assumption which characterizes conventional HMM networks i.e. that of successive observations being independent, can be avoided in hybrid HMM-ANN structures in which the Artificial Neural Network part is designed to provide observation transition probabilities. (Bourlard and Morgan 1998). Here there is no need to i) assume that observations are represented by mixtures of continuous densities, ii) compute observation probabilities using a complete training set of data; these are "global" and do not reflect accurately the short term statistical properties of training data, or iii) assume that successive observations are independent. In contrast multiple inputs to ANN, from a range of time steps, can prime the network to learn about possible correlations that exist between these inputs. However the training complexity characteristic of such hybrid structures are particularly high since for every HMM model re-estimation attempt ANN structure and parameters should be optimized. Note that the classification performance of HMM-ANN schemes can be lower than that offered by HMM-D (Bourlard et al.1995).

Of interest here are extensions and improvements of the Discrete HMM structure of figure 1 into forms and structures that i) can support multiple and often interdependent inputs (feature signals), ii) are modular, iii) can be used as building blocks in designing hierarchical classification systems and also iv) have the capability to provide a dynamically changing with time view and understanding of the importance of each input feature with respect to other input features, within the context of maximizing classification performance. Two new such schemes are presented in the following sections.

HMM-D

In applications where, at a given time instant, observations from multiple sources are available i.e. in the case of multiple input signals, the conventional HMM-D classification approach highlighted in figure 1 can be used in two ways.

First, a Vector Quantizer (VQ) can be employed to represent an input vector (set of observations) with one of the vectors that populate a finite size codebook (Gray 1984). Sequences of quantized vector indices are then modeled by a single HMM-D structure. This approach may initially look relatively simple and therefore attractive. However, since the VQ process involves designing (training) a codebook, HMM parameters must be re-estimated when a new feature is added to the input vector. Moreover, an inaccurate VQ design can degrade significantly system classification performance and in many cases acceptable VQ accuracy is obtained from prohibitively large size, in terms of complexity, codebooks.

An alternative method can be also formulated when input feature signals are assumed to be independent (Lee 1989). In this case a single HMM classification process is designed and employed for each input discrete (quantized) feature signal or data stream. Thus when the observation values $\{o_t^{(1)},o_t^{(2)},...,o_t^{(C)}\}$ taken from *C* feature signals are available at a given time *t*, the system employs *C* HMM classifiers in parallel i.e. there are C models per class, and the total likelihood probability $P(O|\ \lambda^p)$ is given as:

$$P(O|\lambda^p)=\prod_{i=1}^{C}P(O^{(i)}|\lambda_i^p) \quad , \quad p=1,\ 2,..,P \qquad (2)$$

This "multi-HMM-D" process (noted here after as IM-HMM-D (Chiao and Xydeas 2003)) is based on the assumption that input features are statistically independent and may work well for certain type of applications.

DM-HMM-D1, CONNECTING HIDDEN LAYERS

Consider that *C* features are extracted from the same object at a given time and that each feature produces a sequence of observations $O^{(i)}=\{o_1^{(i)},o_2^{(i)},...,o_T^{(i)}\}$, $i=1,...,C$, over a time period *T*. In the conventional IM-HMM-D system of Figure 2(a), sequences of observations are produced from sequences of hidden states (Viterbi traces) $Q^{(i)}=\{q_1^{(i)},q_2^{(i)},...,q_T^{(i)}\}$, which are "independent" between features. However, dependency between features can be introduced as shown in the DM-HMM-D1 (Chiao and Xydeas 2004) system of Figure 2(b) where states that belong to different features (paths) are now "linked" vertically.

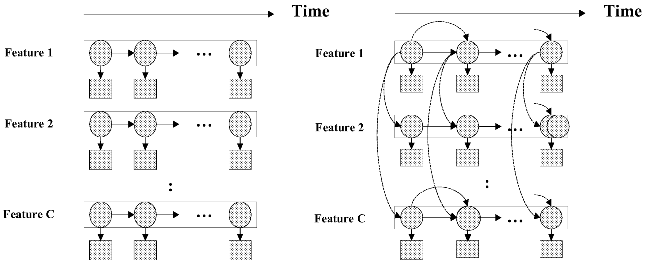


Figure 2 (a) Conventional state transitions across *C* independent HMM paths; (b) Dependency is introduced by linking states across feature paths, DM-HMM-D1.

In this system the resulting likelihood probability is defined as:

$$P^*(Y\mid\lambda)=\log(P(Y\mid\lambda))\approx\sum_{c=1}^{C}\log(w_c(Y)\cdot P(O^{(c)}\mid\lambda_c)) \tag{3}$$

where the weight $w_c(Y)$ is a function of all observations and is attached to the likelihood probability $P\ (O^{(c)}/\lambda_c)$ of the *c*-th feature.

$Y=\{y_1y_2,...,y_t,...,y_T\}$ is a sequence of observation vectors, $y_t=[o_t^{(1)},o_t^{(2)},...,o_t^{(c)},...,o_t^{(C)}]$ is the observation vector at time *t* and $o_t^{(c)}$ is the value of the *c*-th feature (signal) being observed at time *t*. A sequence *Y* of observation vectors can also be represented as $Y=\{O^{(1)},O^{(2)},...,O^{(c)},...O^{(C)}\}$ if we define the sequence of the *c*-th observations as $O^{(c)}=\{\ o_1^{(c)},o_2^{(c)},...,o_t^{(c)},...,o_T^{(c)}\}$.

Furthermore, $w_i(Y)$ effectively represent the importance or otherwise of individual input features within the HMM classification framework.

Note that it is difficult to calculate directly these weights; however, they can be estimated (predetermined) via a "training" process which calculates a *C*×*C* "dependency" matrix *D*.

$$D=\begin{bmatrix} D(N_1,N_1) & D(N_1,N_2) & \cdots & D(N_1,N_C)\\ D(N_2,N_1) & D(N_2,N_2) & \cdots & D(N_2,N_C)\\ \vdots & \vdots & \ddots & \vdots\\ D(N_C,N_1) & D(N_C,N_2) & \cdots & D(N_C,N_C)\end{bmatrix} \tag{4}$$

N_i is the number of states in the *i*-th HMM and $D(N_i,N_j)$ is a $N_i\times N_j$ matrix representing the relationship of connections between the states of the *i*-th and the *j*-th HMMs. The $D(N_i,N_j)$, $1\leq i,j\leq C$ element of matrix D, is described as:

$$D(N_i,N_j)=\begin{bmatrix} d(S_1^{(i)},S_1^{(j)}) & d(S_1^{(i)},S_2^{(j)}) & \cdots & d(S_1^{(i)},S_{N_j}^{(j)})\\ d(S_2^{(i)},S_1^{(j)}) & \cdots & \cdots & d(S_2^{(i)},S_{N_j}^{(j)})\\ \vdots & & \ddots & \vdots\\ d(S_{N_i}^{(i)},S_1^{(j)}) & d(S_{N_i}^{(i)},S_2^{(j)}) & \cdots & d(S_{N_i}^{(i)},S_{N_j}^{(j)})\end{bmatrix} \tag{5}$$

The joint probability of the *u*-th state in the *i*-th model and the *v*-th state in the *j*-th model, is noted as $d(S_u^{(i)},S_v^{(j)})$, and is calculated by counting Viterbi traces (Viterbi 1967) through all *K* training data streams, Note that $1\leq u\leq N_i$ and $1\leq v\leq N_j$. These probabilities are calculated using the following Equations where $q_{k,t}^{(i)}$ indicates the state of the *i*-th observation in the *k*-th data stream at time *t*. T_k is the length of the kth data stream.

$$d(S_u^{(i)},S_v^{(j)})=\frac{\sum_{k=1}^{K}\sum_{t=1}^{T_k}h(q_{k,t}^{(i)},q_{k,t}^{(j)},S_u^{(i)},S_v^{(j)})}{\sum_{k=1}^{K}T_k}\quad ;i\neq j$$

$$d(S_u^{(i)},S_v^{(j)})=\frac{\sum_{k=1}^{K}\sum_{t=1}^{T_k-1}h(q_{k,t}^{(i)},q_{k,t+1}^{(j)},S_u^{(i)},S_v^{(j)})}{\sum_{k=1}^{K}T_k-1}\quad ;i=j \tag{6}$$

Furthermore the counting function *h(a,b,c,d)* is equal to one if and only if the conditions *{a=c}* and *{b=d}* exit, otherwise it is zero.

Furthermore relationship (3) can be developed as

$$\begin{aligned} P^*(Y|\lambda) &\approx \sum_{c=1}^{C}\log(w_c(Y)\cdot P(O^{(c)}\mid\lambda_c))\\ &=\sum_{c=1}^{C}\log(P(O^{(c)}|\lambda_c))+\sum_{c=1}^{C}\log(w_c(Y))\\ &=\sum_{c=1}^{C}\log(P(O^{(c)}|\lambda_c))+\log(w_1(Y))+\log(w_2(Y))+...+\log(w_C(Y))\\ &=P^*_{indep}+\log(w_1(Y)\cdot w_2(Y)\cdots w_C(Y))\\ &=P^*_{indep}+\log(\prod_{c=1}^{C}\log(w_c(Y)))\end{aligned} \tag{7}$$

Assuming that conditional independence exists between sequences of hidden states $Q^{(i)}$ the last expression can be written as

$$
\begin{aligned}
P^{*}(Y \mid \lambda) &\approx P^{*}_{indep} + \log(\prod_{c=1}^{C} w_c(Y)) \\
&= P^{*}_{indep} + \log(\prod_{c=1}^{C}\prod_{t=1}^{T-1} d(q_t^{(c)}, q_{t+1}^{(c)})) + \log((\prod_{i,j=1,i\neq j}^{C}\prod_{t=1}^{T} d(q_t^{(i)}, q_t^{(j)}))) \\
&= P^{*}_{indep} + P^{*}_{dep}
\end{aligned}
\tag{8}
$$

where probabilities $d(q_t^{(i)},q_t^{(j)})$ are obtained from the predefined matrix D. Note that Equations (2) and (3) are the same when the weights w_c are equal to one, i.e. under the assumption that input features are statistically independent. the final maximum likelihood probability expression obtained in this case can be separated into two parts i.e. an independent and a dependent part. Thus the independent probability part P_{indep} is exactly the same as that computed in a conventional IM-HMM-D framework; on the other hand, the dependent probability P_{dep} accounts for the possible relationships that may exist amongst different input features. Figure 3 illustrates diagrammatically the concepts underpinning the IM-HMM-D and DM-HMM-D1 schemes.

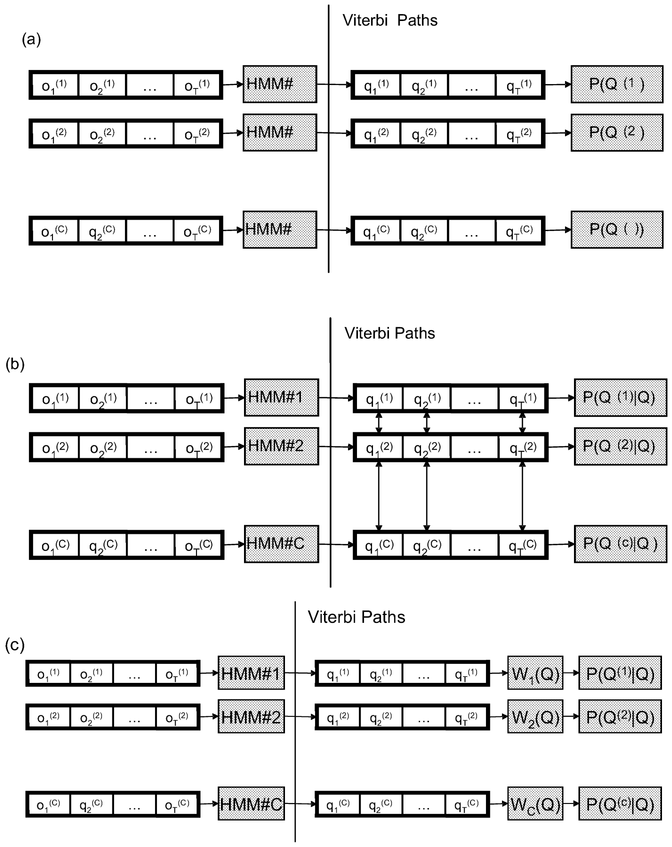


Figure 3: (a) IM-HMM-D, (b) DM-HMM-D1 with states linked vertically across feature streams, (c) equivalent DM-HMM-D1structure with weights operating on every feature stream.

In addition, this formulation of the HMM network not only accounts for possible interdependencies between input feature but can easily lead to an automatic mechanism for the selection /identification of important input features.

DM-HMM-D2, CONNECTING OBSERVATION SEQUENCES

The same broad objectives i.e. i) exploit any interdependence that may exist between input feature data streams and ii) develop the capability to provide a dynamically changing with time view and understanding of the importance of each input feature with respect to other input features, within the context of maximizing classification performance, can be also achieved by linking vertically observation sequences, see figure 4.

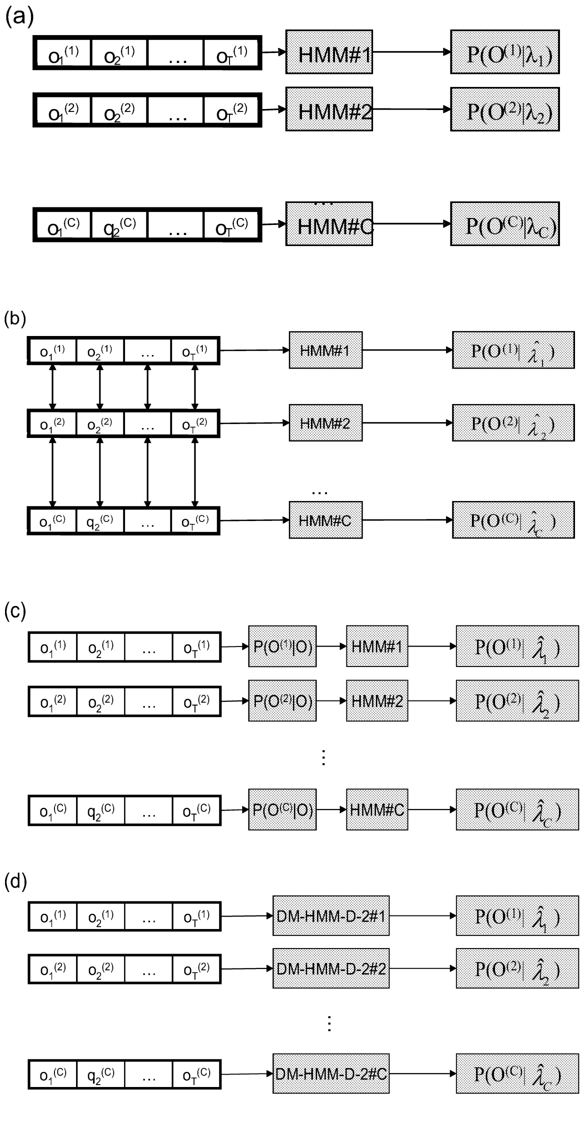


Figure 4: (a) conventional IM-HMM-D, (b) (c) and (d) DM-HMM-D equivalent structures.

Figure 4 (a) shows the conventional IM-HMM-D model framework. Recall that the output probability of each model $P(O^{(i)}|\lambda_i)$ is the same as $P(Q^{(i)})$ shown in Figure 3 (a). Thus Equation 9 provides the conditional probability of the ith observation sequence being produced by the ith model, where $O^{(i)}$ and $\{O^{(1)},O^{(2)},...,O^{(c)},...,O^{(C)}\}$, $c\neq i$ are independent to each other. Note that conditional independence states that if $O^{(i)}$ tells us nothing more about $\{O^{(1)},O^{(2)},...,O^{(c)},...,O^{(C)}\}$, $c\neq i$ than we already know , given λ_i, then $P(O^{(i)}|\lambda_i,Y)$ can be written as:

$$(P(O^{(i)} \mid \lambda_i, Y) = P(O^{(i)} \mid \lambda_i) \qquad (9)$$

where $O^{(i)}=\{o_1^{(i)},...,o_T^{(i)}\}$ is the observation sequence of the i-th feature and $Y=\{O^{(1)},O^{(2)},...,O^{(c)},...,O^{(C)}\}$, $c\neq i$ are the observation sequences of the remaining features. Thus in IM-HMM-D the final likelihood probability is

$$P(O \mid \lambda) = \prod_{i=1}^{C} P(O^{(i)} \mid \lambda_i)$$

In Figure 4 (b), different sequences of observations are considered to be "linked" in a vertical manner by assuming that a weighting function is introduced in prior to each model. The output probability of the i-th model is now written as $p(O^{(i)} \mid \hat{\lambda}_i) \cdot w_i(O)$, where $\hat{\lambda}_i$ is the new HMM parameter set for the i-th feature. In this case

$$P(O \mid \hat{\lambda})' = \prod_{i=1}^{C} (P(O^{(i)} \mid \hat{\lambda}_i) \cdot w_i(O)) \qquad (10)$$

where $w_i(O)$ is a function of both $O^{(i)}$ and Y (note that $Y=\{O^{(1)},O^{(2)},...,O^{(c)},...,O^{(C)}\}$, $c\neq i$). These weights are designed to be the conditional probability of $O^{(i)}$ given Y i.e. the probability of the observation sequence of the i-th feature given the observation sequences of the remaining features, thus

$$w_i(O) = p(O^{(i)} \mid Y) \qquad (11)$$

The system shown in Figure 4(b) now takes the equivalent form of Figure 4(c) which can be also depicted as in Figure 4 (d). This new Multi-HMM model structure is named as DM-HMM-D2 (Xydeas et al. 2006). Since the weight function $w_i(O)$ and the conventional HMM structure are now effectively combined, see figure 4(d), the HMM training (network design) and testing (classification) procedures must be adjusted appropriately. Now,

$w_i(O) = p(O^{(i)} \mid Y)$ can be rewritten as

$$P(O^{(i)} \mid Y) = \prod_{t=1}^{T} P(o_t^{(i)} \mid y_t) \qquad (12)$$

which is also the result of multiplying all probabilities that the Viterbi path passes through, hence

$$\begin{aligned} P(O^{(i)} \mid \lambda_i) &= \pi_{q_1}^{(i)} \cdot b_{q_1}^{(i)}(o_1^{(i)}) \cdot a_{q_1 q_2}^{(i)} \cdot b_{q_2}^{(i)}(o_2^{(i)}) \cdots a_{q_{T-1} q_T}^{(i)} \cdot b_{q_T}^{(i)}(o_T^{(i)}) \\ &= \pi_{q_1}^{(i)} \cdot b_{q_1}^{(i)}(o_1^{(i)}) \cdot \prod_{t=2}^{T} a_{q_{t-1} q_t}^{(i)} \cdot b_{q_t}^{(i)}(o_t^{(i)}) \\ &= f(\pi^{(i)}, a^{(i)}) \cdot \prod_{t=1}^{T} b_{q_t}^{(i)}(o_t^{(i)}) \end{aligned} \qquad (13)$$

and the conditional probability $p(O^{(i)} \mid \hat{\lambda}_i)'$ can be rewritten as:

$$\begin{aligned} P(O^{(i)} \mid \hat{\lambda}_i)' &= P(O^{(i)} \mid \hat{\lambda}_i) \cdot P(O^{(i)} \mid Y) \\ &= f(\pi^{(i)}, a^{(i)}) \cdot \prod_{t=1}^{T} b_{q_t}^{(i)}(o_t^{(i)}) \cdot \prod_{t=1}^{T} p(o_t^{(i)} \mid y_t) \\ &= f(\pi^{(i)}, a^{(i)}) \cdot \prod_{t=1}^{T} b_{q_t}^{(i)}(o_t^{(i)}) p(o_t^{(i)} \mid y_t) \end{aligned} \qquad (14)$$

where the product terms represent the transitional probabilities of the new model, i.e.

$$b_j^{(i)}(o_{(k)}^{(i)})' = b_j^{(i)}(o_{(k)}^{(i)}) \quad p(o_{(k)}^{(i)} \mid y_{(k)}) \qquad (15)$$

It can be seen that the conditional independent probability $P(O^{(i)}|Y)$ will only affect observation transition probability $\{b_j^{(i)}(o_{(k)}^{(i)})\}$. Therefore DM-HMM-D2 can be implemented by replacing $\{b_j^{(i)}(o_{(k)}^{(i)})\}$ with the probability $\{b_j^{(i)}(o_{(k)}^{(i)})'\}$ at each time step (k).

$\{b_j^{(i)}(o_{(k)}^{(i)})'\}$ is calculated using Equation (15) with the help of a pre-defined (during the training procedure) "dependency" codebook that contains $p(o_{(k)}^{(i)}|y_{(k)})$ estimates. In particular, $p(o_{(k)}^{(i)}|y_{(k)})$ estimates are obtained using:

$$\begin{aligned} p(o_{(k)}^{(i)} \big| y_{(k)}) &= p(o_{(k)}^{(i)} \mid (\{o_{(k)}^{(1)}, o_{(k)}^{(2)}, \cdots, o_{(k)}^{(z)}, \cdots, o_{(k)}^{(m)}\})) = \frac{p(o_{(k)}^{(i)}, y_{(k)})}{p(y_{(k)})} \\ &= \frac{\sum_{k=1}^{K} \sum_{(k)'=1}^{T_k} h(O_{k,(k)'}, O_{(k)}) / \sum_{k=1}^{K} T_k}{\sum_{k=1}^{K} \sum_{(k)'=1}^{T_k} h(U_{k,(k)'}(i), V_{(k)}(i)) / \sum_{k=1}^{K} T_k} = \frac{\sum_{k=1}^{K} \sum_{(k)'=1}^{T_k} h(O_{k,(k)'}, O_{(k)})}{\sum_{k=1}^{K} \sum_{(k)'=1}^{T_k} h(U_{k,(k)'}(i), V_{(k)}(i))}, z \neq i \end{aligned} \qquad (16)$$

where $U_{k,(k)'}(i)=\{o_{k,(k)'}^{(1)},o_{k,(k)'}^{(2)},...,o_{k,(k)'}^{(z)},...,o_{k,(k)'}^{(m)}\}$ and $V_{(k)}(i)=\{o_{(k)}^{(1)},o_{(k)}^{(2)},...,o_{(k)}^{(z)},...,o_{(k)}^{(m)}\}$ with $z \neq i$ are calculated as the expected number of times in observing $V_{(k)}(i)$ for all $U_{k,(k)'}(i)$ in K training data sets, $k=\{1,2,...,K\}$. The counting function $h(a,b)$ is equal to one if and only if $\{a=b\}$, otherwise its value is zero.

APPLICATION 1: EEG BASED CLASSIFICATION OF PAIN AND NON-PAIN CONDITIONS

EEG signals represent "average" electrical activities (i.e. sum of the excitatory and inhibitory postsynaptic potentials) of thousands of neurons over a portion of the brain and almost entirely from the upper layers of the cerebral cortex. However, despite this "averaging" process that characterizes the EEG representation of brain activity, EEG signals are very useful in clinical diagnosis and medical research as they provide an observation access to physiological, pathological, and even psychological/mental states of human beings. Of course this statement implies the collective knowledge and understanding of temporal EEG pattern characteristics, of spatial or topographical EEG features and in general of empirical observations and associations between EEG signals and physiological /pathological conditions.

Whereas EEG signals provide an important and often rich source of information, multi-channel EEG recordings generate large amounts of data whose meaningful analysis and interpretation is a difficult task (Baltas et al 2002).

This section presents the application of the D-HMM-D2 classification methodology, in the processing of 64-channel EEG signals, with the objective of recognizing "pain" or "no pain" conditions automatically from the signals and of course in agreement with the actual experience of subjects.

Thus this work can be viewed as a contribution towards the development of human-machine interface systems that are designed to automatically detect and classify pain states (Stergioulaset al. 2002).

In addition, the system can be also used to provide dynamic (time changing) information on the contribution that different input EEG channels make towards an overall accurate classification system performance.

Data Acquisition and Experimentation

Repeated heat stimuli in the form of laser pulses were delivered to the right forearm of a subject by CO$_2$ laser in a controlled manner. The duration of each pulse is set to 150ms, and each such stimulus is repeated at regular intervals of 10s (epochs). Each EEG continuous recording included 61 stimuli. Note that EEG (channel) signal responses to the first stimulus, were routinely discarded, as they were considerably higher in amplitude due to an element of "surprise" that is often exhibited by the subject and associated artefacts in the EEGs. Thus, 60 stimuli were taken into account for each recording and nine continuous EEG data files were produced from a healthy female subject. These data files were recorded on three different days.

Recordings were made using a 64-electrode cap (see Figure 5) with 62 head electrodes while two face electrodes (vEOG and hEOG) were used to monitor artefacts from eye movement. EEG signals were recorded at a sampling frequency of 500Hz, with a gain of 500 (150 for the EOG channels). These signals were also band-pass filtered at the range of 0.15Hz to 30Hz.

Figure 5: EEG cap in its physical form mounted on the scalp of a subject

In the pain classification experiments discussed in this section, a total of 710 available segments (355 pain segments and 355 no pain segments) were used for training the classification HMM networks. Each of these segments was obtained from the first eight (out of nine available data files) "training" EEG files by taking 500 sample values located before the occurrence of each heat stimuli, as representing a period of "no pain" signals and 500 sample values located after the heat stimuli, as representing "pain" signal periods. Furthermore 34 such segments were taken from the 9th file for representing each class in the testing data set.

Note that input EEG signals were also pre-processed, as required by the following Discrete HMM classification systems. Thus EEG sample values were scalar quantized at different resolution with M=100, 50, 20, or 10. Notice that M is also equal to the number of different discrete observation values that can produced from a network state.

The overall experimental procedure is shown in figure 6 and involves the network training and system classification performance phases for the conventional IM-HMM-D structure and the DM-HMM-D2 system. A useful by-product of the second technique is the "weight" information that is attached by DM-HMM-D2 to each input signal.

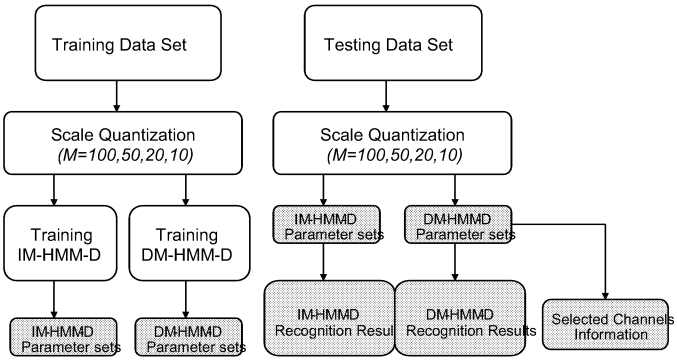


Figure 6: Design and evaluation Procedures used in EEG classification experiments

Table 1 provides a comparative list of the classification performance results obtained from the IM-HMM-D and DM-HMM-D2 schemes.

In the IM-HMM-D experiments, models operating with different M values (i.e. *M*=100, 50, 20 and 10) result in a similar performance and provide a relatively high recognition accuracy with respect to the "no pain" condition (i.e. of the order of 91 to 97%) and a definitely low recognition accuracy for the "pain" condition (i.e. of the order of 41 to 53%). It is also clear that given a model with a relatively small number of hidden states ($N=10$ in this case), an increase in the value of M does not lead to better performance. In other words, as the value of M increases and thus variability in the "shape" of possible input patterns increases, a gain in classification accuracy (as a result the increase in the amount of input information) can only be achieved by a corresponding increase in the value of N (network size) and hence by an increase in the complexity of the HMM network. Thus given a network value N there is an appropriate range (M_R) of M values for which classification performance is maximised. When M<< (M_R) there is excessive quantisation noise that adversely affects classification performance. In contrast to this, when M>>(M_R) the complexity/variability of the input data is excessively large for the given network (and value of N) to accurately model/encode the input data. Furthermore there are no strict procedures, other than experimental investigations, that can be used to determine the appropriate values for N and M, in terms of maximising the classification performance of the system. Nevertheless as a rule of thumb $M \leq 10N$ (Chiao 2005)

As mentioned previously the IM-HMM-D system exhibits considerable variability in classification accuracy between the two classes. When performance is calculated as an average value using both classes, the system delivers 69.11765% with M=20; and maximum 72.0588% with M=10. Note that these classification performance figures are comparable to the 70% accuracy results obtained from M. Baltas (Baltas et al 2002) using a hybrid Learning Vector Quantisation (LVQ) network which was designed (i.e. trained) and tested while employing the same EEG data files and "pain" ,"no pain" signal segments.

Considerably higher classification accuracy rates are obtained by applying the DM-HMM-D2 system, see Table 1. This improved performance is obtained for all values of M. and the system operates best (with a performance in excess of 92%) with M values in the region of 20 to 50. Note that when several different models provide similar classification performance, the structure with lower values for M and N is preferred due to lower system complexity. Thus the model with M=20 and N=10 is used below to obtain an "usage rate" for each EEG channel and hence an indication of the significance of each channel.

Scale Model	100		50		20		10	
	Pain	Cont.	Pain	Cont.	Pain	Cont.	Pain	Cont.
IM-HMM-D	41.1765%	97.0588%	40.5882%	97.0588%	47.0588%	91.1765%	52.9412%	91.1765%
DM-HMM-D	91.1765%	97.0588%	97.0588%	94.1176%	97.0588%	94.1176%	55.8824%	92.9412%

Table 1: Classification results for the IM-HMM-D and DM-HMM-D2 systems. Classification performance is expressed as the percentage of the testing EEG segments that were classified correctly

In this case a threshold value $W_{threshold}$ is defined and applied to $p(o_t^{(i)}|y_t)$, $t=1,...,T$ and $i=1,...,C$. $w_i(O= p(o_t^{(i)}|y_t)$ values which are less or equal to $W_{threshold}$ are considered to be insignificant and indicate that the contribution to the overall classification probability $p(O^{(i)} | \hat{\lambda}_i)$ of the corresponding "ith"feature at time "t", is also insignificant. Thus by applying the above threshold based criterion, features are classified as "active" or "non-active" and a feature "usage rate" can be therefore estimated.

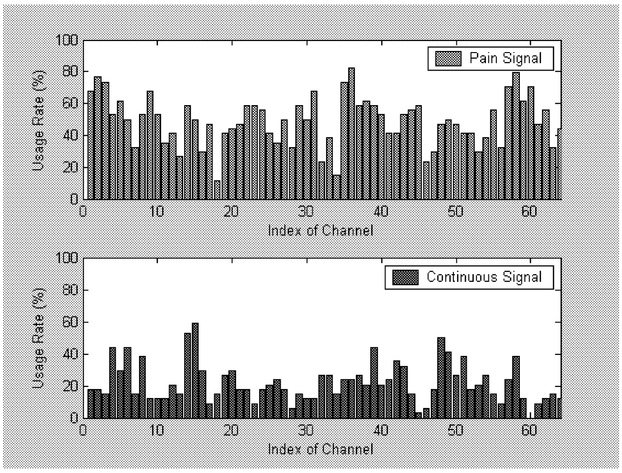


Figure 7.3: Channel usage rates in DM-HMM-D2 pain, no-pain classification

Figure 7 shows the usage rate of each input channel(feature) as calculated from a classification experiment with DM-HMM-D2 operating on a total of 68 EEG segments (34 segments used for each class). In the experiment, $W_{threshold} =10^{-5}$ Certain channels (for example, Channels 4, 6, 15, 16, 39, 48, 49, 51, and 58) are heavily involved in the classification of both pain and no pain conditions. In general, this input channel categorization methodology can be particularly useful to researchers interested in the reduction of the number of input channels (features) presented at the input of a classifier with a carefully controlled effect on classification performance.

Application 2: Classification and Prediction of LPC Speech Model parameters in VoIP

In order to deliver real time, high quality voice services, VoIP system designers must tackle the packet-loss problems that are inherent in packet-based networks. To combat the inevitable speech quality deterioration resulting from the loss of transmitted packets of speech information, techniques that provide estimates of the lost information that is needed by the speech recovery process are of considerable interest. Here it is assumed that the next generation of VoIP systems will employ efficient parametric speech coders which are most likely to be of the Analysis-by-Synthesis LPC type, since all the currently established speech coding standard algorithms operating at bit rates in the range of 2.4 Kbits/s to 13 Kbits/s are LPC based. This means that a substantial part of the transmitted speech information will be in the form of LPC parameters and when transmitted packets are missing, estimates of these parameters should be formed at the receiving end and used by the speech recovery (decoding) process. Another assumption made is that LPC information is efficiently quantized using the Split Matrix Quantization (SMQ) method (Xydeas and Papanastasiou 1999).

The HHM based estimation of missing LPC filter information described here and in (Chiao et al. 2005) involved sets of ten LSP coefficients. These sets are calculated originally from the analysis of successive 20 msecs speech segments and LPC coefficients are transformed to LSP coefficients (an equivalent vocal tract parametric representation form that is more appropriate for quantization), prior to applying SMQ. Furthermore in these experiments SMQ operates over four 20msecs frames (one SMQ frame is equal to four LSP frames, i.e. 80 msecs), that is over four sets of ten LSP coefficients, in order to produce one set of ten SMQ indexes (LSP_i , i=1,…,10). Also, the LPC information that is contained within each transmitted packet corresponds to an 80 msecs speech segment (i.e. one SMQ frame).

Now, given that the current packet is missing and can not therefore be used in the speech synthesis process, an HMM process that operates on previously received SMQ information, can be used to determine a "most-likelihood" estimate for the missing SMQ indices. In general, the HMM period of observation time T_{ob}, consists of current "c" (missing) and previous "p" (received) SMQ frames, is $T_p=(c+p)T_{pa}$, where $c<<p$. In these experiments c=1, p=3 and T_{pa}=80 msecs.

In the case of a missing packet, the LSP recovery/estimation process operates on four successive SMQ frames with each frame represented by ten SMQ quantization index values and the last frame being that of the missing packet. Thus the estimation process employs a "bank" of ten HMM$_i$ models, i.e. i=1,..,10 with the "i-th" model operating on the observation sequence $O^{(i)}=\{o_1, o_2, o_3, o_4\}^{(i)}$, see figure 8.

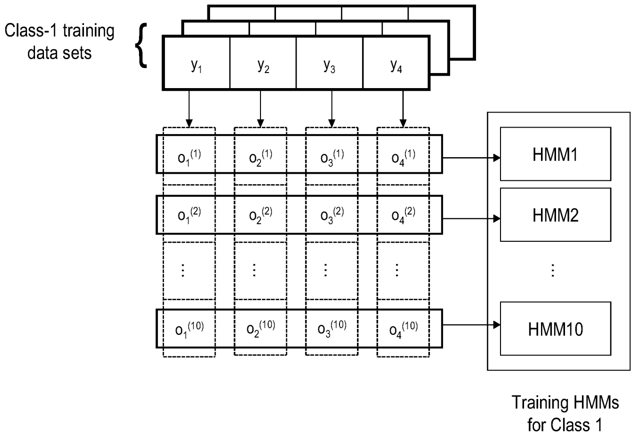


Figure 8. Training HMMs on a per class basis.

Alternatively, this bank of HMMs operates on a sequence of four observation (column) vectors $y_t=\{o_t^{(1)},o_t^{(2)},\ldots,o_t^{(10)}\}$ where t =1,..4 are the time indices of SMQ frames. However since the last (4th) observation vector is missing, only the y_1, y_2 and y_3 vectors are available to the input of the HMM bank, thus over the 320 msecs observation period and the original sixteen LPC vectors of ten coefficients each, which were quantised to four SMQ vectors of ten coefficients each, only the first three of the SMQ vectors are available to the LSP recovery process, i.e.

$$\begin{bmatrix} o_1^{(1)} & o_2^{(1)} & o_3^{(1)} \\ o_1^{(2)} & o_2^{(2)} & o_3^{(2)} \\ \vdots & \vdots & \vdots \\ o_1^{(10)} & o_2^{(10)} & o_3^{(10)} \end{bmatrix} = Y = \begin{bmatrix} y_1 & y_2 & y_3 \end{bmatrix} \qquad (17)$$

Now, having y_1, y_2 and y_3 and using y_4^* to indicate an estimate of the missing SMQ vector y_4, this estimate can be selected such that the likelihood probability of observation $Y^*=\{y_1, y_2, y_3, y_4^*\}$, given an HMM model λ, i.e. $P(Y^*=\{y_1, y_2, y_3, y_4^*\})|\lambda)$ is maximum.

This effectively means that the HMM bank is designed i.e. trained to classify input patterns of four SMQ vectors $\{y_1, y_2, y_3, y_4\}$ to one of λ classes. Then given the resulting λ HMM bank models and an incomplete observation vector sequence $\{y_1, y_2, y_3\}$, the system defines y_4^* so that $P(Y^*=\{y_1, y_2, y_3, y_4^*\})|\lambda)$ is maximum over all λ classes.

Of course each of these classes should represent clusters of "similar" four-vector SMQ "sequences" Y whose characteristics are captured by the corresponding bank of HMM models. This concept of having clusters of "similar" Y sequences of SMQ vectors can be easily accepted due to the structure imposed on the speech signals in general and LSP tracks in particular by language rules and human speech production mechanism constrains.

Note that a SMQ frame size of 80 msecs and T_{ob} of 320 msecs were selected to reflect phoneme/syllabic durations. The voiced-unvoiced nature of the speech signal was also selected as the clustering criterion, an assumption that results however in significant variability between Y sequences belonging to the same class but, at the same time, leads to a small number of classes, λ=7. Also note that voiced/unvoiced classifications are produced at the output of a Voiced Activity Detection (VAD) process operating on a 20 msecs frame basis and thus Y sequences defined over T_{ob} = 320msecs are classified into the λ=7 classes using 16 voiced/unvoiced flags (320/20=16). The classes are 1) voiced, 2) unvoiced, 3) voiced to unvoiced, 4) unvoiced to voiced, 5) voiced to unvoiced to voiced, 6) unvoiced to voiced to unvoiced and 7) other.

Now given seven classes and ten different HMM networks in the system bank, see figure 1, the probability of observing $O^{(i)}$ given the the j-th class i-th HMM model is $P(O^{(i)}|\lambda_{j,i})$, with $i=1,\ldots,10$ and $j= 1,\ldots,7$. Then the total probability $P(Y|\lambda)$ over the system bank can be maximized using:

(1) $P(Y|\lambda)=\max\{P(O^{(1)}|\lambda_{1,1})\times P(O^{(2)}|\lambda_{1,2})\times\ldots\times P(O^{(10)}|\lambda_{1,10}) ,P(O^{(1)}|\lambda_{2,1}), P(O^{(2)}|\lambda_{2,2}),\ldots, P(O^{(10)}|\lambda_{2,10}) ,\ldots., P(O^{(1)}|\lambda_{7,1}), P(O^{(2)}|\lambda_{7,2}),\ldots, P(O^{(10)}|\lambda_{7,10}) \}$

or

(2) $P(Y|\lambda)=\max\{P(O^{(1)}|\lambda_{1,1}),P(O^{(1)}|\lambda_{2,1}),\ldots, P(O^{(1)}|\lambda_{7,1}) \} \times\max\{ P(O^{(2)}|\lambda_{1,2}), P(O^{(2)}|\lambda_{2,2}),\ldots,P(O^{(2)}|\lambda_{7,2}) \} \times\ldots \times\max\{ P(O^{(10)}|\lambda_{1,10}), P(O^{(10)}|\lambda_{2,10}),\ldots, P(O^{(10)}|\lambda_{7,10}) \}$

That is, $P(Y|\lambda)$ is maximized by defining (1) the largest product of 10 $P(O^{(i)}|\lambda_{j,i})$ probabilities, with each product defined within a given class i.e. the value of j is fixed or (2) each of the 10 probability terms (i=1,.,10) which form the product that defines $P(Y|\lambda)$ maximum is itself the maximum $P(O^{(i)}|\lambda_{j,i})$ value across j=1,., λ with the value of i fixed.

The performance of the proposed SMQ index estimation (LSP recovery) methodology has been evaluated experimentally via computer simulation with the HMM models bank system implemented using two different approaches, that is IM-HMM-D and DM-HMM-D2 . Furthermore system performance was evaluated while processing input files (e.g. “Hello operator” file yielding 307 SMQ vectors), which were not included in the generation of the data used in the HMM model training process.

In these experiments missing packets/SMQ vectors were “introduced” at regular intervals, in the sequence of SMQ vectors generated for each of the test files. That is, “*m*” SMQ vectors were assumed as missing every “*n*” SMQ vectors, in the test sequence, for example m=1 and n=8. In addition it has been assumed that the transmitted LSP parameters are those of a Pitch Synchronous Prototype Interpolation Manchester coder (Zafiropoulos and Xydeas 2003) operating at 2.4 Kbits/sec and thus voiced/unvoiced flags are also available at the receiver.

The quality of the recovered speech signal at the output of the Pitch Synchronous Prototype Interpolation Manchester decoder was measured using the objective Perceptual Evaluation of Speech Quality (PESQ) method (Ordas and Fox 2004) that returns a quality score in the region of 0 to 5. Table 2 shows the PESQ scores obtained by using the IM-HMM-D and DM-HMM-D2 to predict the missing speech LSP vectors.

	IM-HMM-D		DM-HMM-D2		Encoder/Decoder Performance (with missing vectors)
Probability Selection type	Method 1	Method 2	Method 1	Method 2	
PESQ Score (Hello)	1.804188	1.808352	1.89119	1.995804	1.946306

Table 2: PESQ Scores of the predict results using HMMs

The encoder/decoder performance is the reference performance that is obtained by reproducing missing speech LSP vectors (y_4) using previously received vectors (y_3). These results are typical to those obtained from several experiments and show that for the “Hello” input speech file DM-HMM-D2 provides higher scores than IM-HMM-D. A PESQ score of 1.995804 is obtained via applying the DM-HMM-D2 process and Method 2 Thus experiments indicated a slightly higher PESQ score than that (i.e.1.946306) of the case which replaces the missing speech LSP parameters with the previously received SMQ vector y_3.

When looking into the HMM parameter sets, the diagonal values of the matrix A, where $A=\{a_{ij}\}$ represents the hidden state probabilities, $1\leq i\leq N$ and N is the number of hidden states, are found to be significantly higher than the rest of the a.

This implies that when using HMM to estimate the next observation (o_4^*), o_4^* is more likely to be generated by the hidden state s_j which previously produced the observation o_3. In this case, o_4^* is estimated by selecting the o_m with the highest observation probability $b_j(o_m)$, where $1\leq m\leq M$ is the range of possible observation and M is the maximum scalar quantized value i.e. the estimated o_4^* is highly likely to assume the same value given a hidden state s_j.

This situation arises from the small number of classes λ i.e. seven, used in these experiments to represent/model the various Y sequences of SMQ indexes. As it was mentioned earlier, each of these classes should represent clusters of “similar” four-vector SMQ “sequences” Y and the current voiced/unvoiced based criterion for generating these classes is introducing an excessive degree of interclass variability. A different clustering criterion that relates better to the “shape“ characteristics of the SMQ sequences and thus offers enhanced similarity between sequences belonging to the same class, is therefore required.

DISCUSSION

Both DM-HMM-D1 and DM-HMM-D2 schemes are generic and offer significant performance advantages over conventional HMM-D and IM –HHH-D modeling structures. They are modular and also provide a dynamic “view” of the importance or otherwise of input feature signals (streams of data). Also they are ideally suited to operate on real time signals and when these signals are non stationary, both methods can be integrated optimally with signal segmentation processes so that performance is maximized (Chiao and Xydeas 2003) (Chiao and Xydeas 2004).

In addition both schemes easily render themselves in the creation of hierarchical classification / prediction structures made of several HMM based layers of modeling networks which progressively fuse incoming information to higher level forms of “objects”, or “events”, within the context of specific applications and situations.

The next step forward in the further development of these schemes must be focused on work that allows these probabilistic networks to adapt their structures on line in a computationally efficient manner, while performance is maximized.

REFERENCES

Baltas, E.; D.Bentley; and C. Xydeas, and others, 2002 *“An LVQ Classifier of EEG Coherence Patterns for Pain Detection.”* Proc. Intern. Conference CSNDSP’02, University Press, pp. 248-251

Bengio, Y. 1999. “Markovian Models for Sequential Data” *Neural Computing Surveys*, vol.2: 129-162.

Bourlard, H. and N. Morgan, 1998 *“Hybrid HMM/ANN Systems for Speech Recognition: overview and new directions”* Techn. Report Intern. Comp. Science Institute, Berkeley , CA.

Bourlard, H; Y. Konig; and N. Morgan, 1995, *“REMAP: Recursive Estimation and maximization of a posteriori probabilities in connectionist speech recognition”* Techn. Report , Intern. Computer Science Institute, Berkeley .

Chiao, Shih-Yang and C. Xydeas 2003, **“** *Modeling Behaviors of Players in Competitive Environments”* IEEE, WIC Conference on Intelligent Agent Technology, Halifax, Canada.

Chiao, Shih-Yang; C. Xydeas 2004 *“Using Hierarchical HMMs in Dynamic behavior modeling”* Proc. 7th Intern. Conference on Information Fusion , Fusion2004, Stockholm, pp. 576-582

Chiao S.; C. Xydeas; E. Jones , 2005, *“ Recovery of LSP coefficients in VoIP Systems”* 5th Inern. Conf. On Information Communications and Signal Processing, IEEE, Bangkok , Thailand.

Chiao, Shih-Yang 2005 *“Probabilistic Modeling of*

Behavioral Patterns" PhD Thesis , Lancaster University, UK.

Ghahramani, Z, 1999, "Factorial Hidden Markov Models" *ournal of Machine Learning,* vol. 29, pp. 245-275.

Ghahramani, Z. 2001. "An Introduction to Hidden Markov Models and Bayesian Networks" *International Journal of Pattern Recognition and Artificial Intelligence*, vol.15, no.1: 9-42.

Gray, R. 1984, "Vector Quantization" *IEEE ASSP Magazine*:4-29.

Huang, X.; Lee, K.F.; and Hon, H.W. 1990, "On semi-continuous hidden Markov modeling", *International Conference on Acoustics, Speech and Signal Processing: 689-692.*

Lee, K.F. 1989. *Automatic Speech Recognition: the Development of the SPHINX System, Kluwer Academic Publications.*

Moon, T.K. and Sterling, W.C. 2000, *Mathematical Methods and Algorithms for Signal Processing,* Prentice-Hall, Englewood Cliffs, N.J.

Ordas, P. and B. Fox, 2004 "Perceptual Evaluation of Speech Quality (PESQ)", Technical Report, Microtronix Systems Ltd.

Rabiner, L.R., 1989. "A Tutorial on Hidden Markov Models and Selected Application in Speech Recognition" *Processing of IEEE*, vol.77, no.2: 257-285.

Stergioulas, L.; C. Xydeas; and others, 2002 *"EEG signal analysis ysing average cross-channel coherence"* Proc, 2nd European Medical and Biological Eng. Conference EMBE'02, Vol. 1, pp. 434-436

Viterbi, A.J. 1967. "Error Bounds for Convolutional Codes and an Asymptotically Optimal Decoding Algorithm" *IEEE Trans. Information Theory,* Vol.IT-13:260-269.

Xydeas, C; P. Angelov; S. Chiao; and M. Reoulas, 2006 *" Advances in classification of EEG Signals via Evolving Fuzzy classifiers and Dependent Multiple HMMs"* Computers in Biology and Medicine, Vol. 36, issue 10 pp. 1064- 1083

Xydeas, C. S. and C. Papanastasiou, 1999, "Split Matrix Quantization of LPC parameters", Proceedings of IEEE Transactions in Speech and Audio, Vol. 7, No. 2, pp.113-125, March.

Zafiropoulos, F.; C. Xydeas 2003 *"Model based packet loss*

concealment for AMR Coders" IEEE Intern. Conf. on ASSP, Vol. 1 pp. 112-115

REDUCING COMPLEXITY IN THE SYSTEMATIC CONSTRUCTION OF PETRI NETS MODELS THROUGH GRAPH TRANSFORMATIONS

Carmen-Veronica Bobeanu and Hendrik Van Landeghem

Ghent University, Department Industrial Management, Technologiepark 903,
B-9052 Ghent, Belgium
e-mail: carmen.bobeanu@UGent.be
hendrik.vanlandeghem@ UGent.be

KEYWORDS
Petri Nets, Meta-Modelling, Graph Grammars, Discrete Event Systems

ABSTRACT

This paper addresses the authors' current results to generate customised tools supporting the synthesis of compound Petri nets models of complex discrete event systems. The proposed approach combines the advantages of meta-modelling (to minimise the effort required to construct a tool which accepts refined structures of Petri nets models proposed by the authors) and graph transformation systems (to implement the rules underlying the handling of the set of routing constructs identified in the authors' work). These provide the basis for further automated generation of complex Petri nets models from the appropriate graph grammars.

1. INTRODUCTION

Nowadays, simulation is widely accepted as an indispensable methodology to tackle problems of ever increasing complexity. Although a huge amount of work has been invested in the development of simulation software tools, it is recognized that simulation is still not routinely used in many cases where it can have a significant impact. A major drawback as it appears from the practice of simulation and process modelling is referring to the model specification techniques, which have been, and continue to be, ad hoc. Therefore, much effort is still to be done in this respect to move the adoption and use of simulation forward in the future. This orientation is acknowledged by the remark made in (Vangheluwe et al. 2001) that, "to tackle problems of ever increasing complexity of today's highly competitive systems, modelling and simulation research is shifting from simulation techniques to modelling methodology and technology".

In this perspective, it is the purpose of the research considered in this paper to provide a natural, simple and powerful method for describing and analysing discrete event systems (DES) not far from the industrial engineers' habitual notions of system design and operation. The proposed approach combines the advantages of meta-modelling (to avoid explicit programming of a customised tool which accepts refined structures of Petri nets models proposed by the authors) (Van Landeghem and Bobeanu 2003) and graph transformation systems (to define allowed manipulations of the structures we want to deal with) (Bobeanu and Van Landeghem 2005). This contribution revisits the authors' previous work (Van Landeghem and Bobeanu 2002, Van Landeghem and Bobeanu 2003, Bobeanu et al. 2004) and integrates into a new setting results addressing the modular synthesis of re-usable Petri nets (PN) models of DES with the use of graph transformations (Ehrig and Padberg 2004) to relate the nets in different development stages.

2. A SYSTEMATIC CONSTRUCTION OF MULTICOMPONENT PN MODELS

We recall that the central idea of our research is a refined PN representation in terms of sets and operations on sets, focusing on the main attributes of a composite PN object specification and anticipating the application of Zeigler's theory (Zeigler et al. 2000) in the PN domain. We recall as well the key elements supporting the manipulation of these abstractions introduced in our previous work (Bobeanu and Alla 1998, Bobeanu et al. 2004):

- *primitive/atomic model*: block component with a well-defined interface, that can be seen as a process with an input transition, T_i, supplying the input of its activity and an output transition, T_o, enabling its evacuation and a place P modelling the temporal entities advance in the net (production place) or constraints to be satisfied (synchronisation place);
- *coupling templates*: standardised means to couple building block components covering three aspects:
  i. external input coupling (the rule used for establishing the input port of the coupled model);
  ii. external output coupling (the rule used for establishing the output port of the coupled model);
  iii. internal coupling (the fusion rule of the input port of one component with the output port of the other one).
- *step-by-step procedure for the construction of compound PN-models*: systematic bottom-up construction of PN models using as input information (about e.g. primitive system components, entity flows, routing constructs, etc.) gathered from the top-down system analysis.

One of the new ideas considered in (Van Landeghem and Bobeanu 2002) is that the development of a well-grounded set of coupling templates forces one to partition the entity flows into three distinct types when an implementation of the proposed approach to supply chain modelling is addressed:

1. orders placed by the customer
2. internal orders
3. products.

3. META-MODELLING THE REFINED PETRI NETS FORMALISM

In order to support the above approach, we are developing a modelling tool in the refined formalism proposed in our method. Since the time needed to develop such tools is normally prohibitive, we have chosen for using meta-modelling, as a means to tackle the above problem.

The concepts proposed in our developments addressing the modular synthesis of PN-based simulation models of complex DES are implemented in AToM^3, A Tool for Multi-formalism, Meta-Modelling. As emphasized in (Lara and Vangheluwe 2004), in the meta-modelling approach followed in AToM^3, meta-models are regarded as type graphs (with inheritance) enriched with certain constraints. Some of them (e.g. cardinalities) are visually embedded in the type graph, while others (”well-formedness” rules) are defined using a textual language (Python code).

The main steps in the generation of the Refined Petri Nets (RPN) formalism from a model designed in the Entity-Relationship formalism, using AToM^3 are explained in (Van Landeghem and Bobeanu 2003). Currently we are using a description of the above meta-model in terms of UML Class Diagrams, adopted as the default formalism for generating new formalisms in recent versions of AToM^3.

The window in Figure 1 shows two classes of objects referred by our meta-model (RPN_Place and RPN_Transition, respectively) with a description of their corresponding list of attributes.

We insist on the introduction of particular abstract information supporting the implementation of the extensions to the PN formalism envisaged in our work.

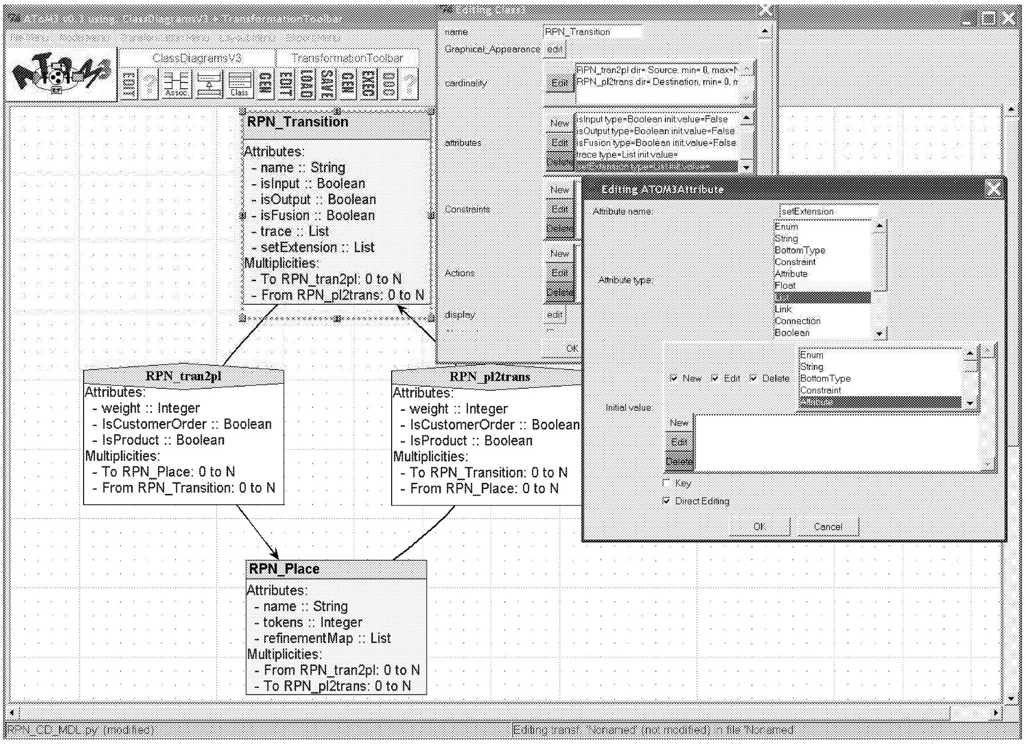
The refinement of the transitions set introduced in our formalism was implemented by providing the RPN_Transition class with three Boolean attributes (isInput, isOutput and IsFusion), allowing the user to specify if the current transition is an input or output port of the model, or the result of applying a transition fusion respectively, when further manipulation of basic model components will be addressed (see Figure 2). The graphical expression of objects in the class RPN_Transition was specified by the property Appearance, where different graphical objects were envisaged for a transition depending if it is input, output or fusion transition. We have also added two Python coded graphical constraints to modify the appearance of a transition object depending if the Boolean variables isInput or isOutput are set to .TRUE. or to .FALSE. (see Figure 2).

Recent work was devoted to capturing supplemental abstract information in the list of class attributes to anticipate further implementation of the coupling mechanism using graph transformations (see Figure 1):

-- The substitution of an input/output transitions set by a representative system is “traced” by providing the RPN_Transition class with an attribute of type List, *setExtension*.

-- A “memory” of the aggregation process of the internal structure of a compound model is kept by providing the RPN_Place class with an attribute of type List, *refinementMap*.

The above information will be of use in the reversion to a detailed description of the overall model.



RPN_Transition
Attributes:
- name :: String
- isInput :: Boolean
- isOutput :: Boolean
- isFusion :: Boolean
- trace :: List
- setExtension :: List
Multiplicities:
- To RPN_tran2pl: 0 to N
- From RPN_pl2trans: 0 to N

RPN_tran2pl
Attributes:
- weight :: Integer
- IsCustomerOrder :: Boolean
- IsProduct :: Boolean
Multiplicities:
- To RPN_Place: 0 to N
- From RPN_Transition: 0 to N

RPN_pl2trans
Attributes:
- weight :: Integer
- IsCustomerOrder :: Boolean
- IsProduct :: Boolean
Multiplicities:
- To RPN_Transition: 0 to N
- From RPN_Place: 0 to N

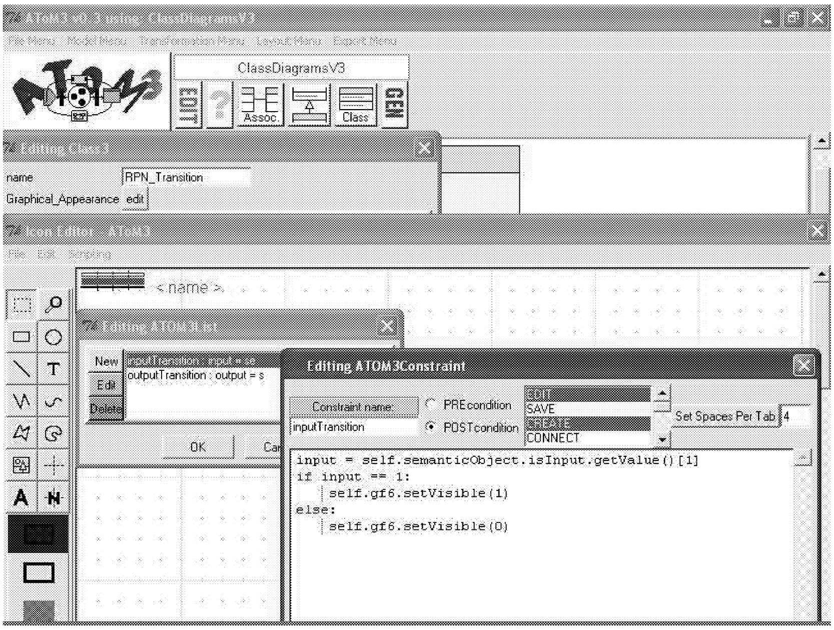
RPN_Place
Attributes:
- name :: String
- tokens :: Integer
- refinementMap :: List
Multiplicities:
- From RPN_tran2pl: 0 to N
- To RPN_pl2trans: 0 to N

name: RPN_Transition
cardinality:
RPN_tran2pl dir= Source, min= 0, max=N
RPN_pl2trans dir= Destination, min= 0, m
attributes:
isInput type=Boolean init.value=False
isOutput type=Boolean init.value=False
isFusion type=Boolean init.value=False
trace type=List init.value=

Editing ATOM3Attribute
Attribute name: setExtension
Attribute type: List
☑ New ☑ Edit ☑ Delete
□ Key
☑ Direct Editing

RPN_CD_MDL.py (modified)
Editing transf 'Nonamed' (not modified) in file 'Nonamed'

Figure 1. Meta-Model for Refined Petri Nets Formalism



name: RPN_Transition
Graphical_Appearance: edit

New
Edit
Delete
outputTransition : output = s
OK

Editing ATOM3Constraint
Constraint name: inputTransition
○ PREcondition
● POSTcondition
EDIT
SAVE
CREATE
CONNECT
Set Spaces Per Tab: 4

```
input = self.semanticObject.isInput.getValue()[1]
if input == 1:
    self.gf6.setVisible(1)
else:
    self.gf6.setVisible(0)
```

Figure 2. Setting Up the Appearance of a Transition Object

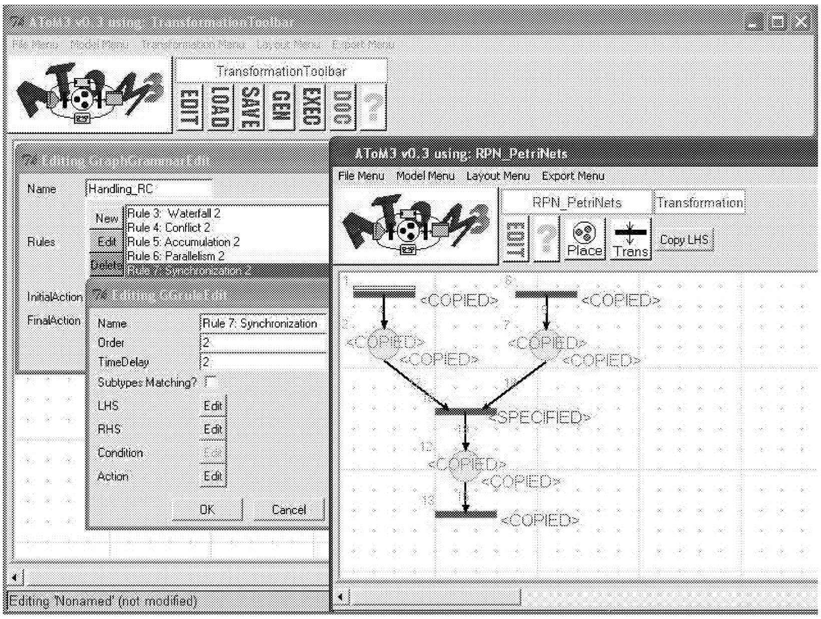
The tool generated in AToM^3 according to the above meta-specifications can be used to build, load and save refined representations of PN models proposed in our approach.
Recent developments were aimed to extend the generated tool's functionality by addressing compound models synthesis. As the meta-model defined in previous step is represented in AtoM^3 as attributed, type graph, these operations were expressed in a natural way using graph grammars (GG).

4. SYNTHESIS OF COMPOUND MODELS THROUGH GRAPH TRANSFORMATIONS

By exploiting the representation of the meta-model defined in Section 3 as attributed, typed graph, our recent developments express the systematic construction proposed in our previous work (Bobeanu et al. 2004) as high-level model in the formal, graphical and intuitive notation of GG. Modelling the handling of the routing constructs (RC) identified in our approach using graph transformation is natural and intuitive and the effects of applying the coupling mechanism can be visually traced.

The Coupling step in the step-by-step procedure proposed in (Bobeanu et al. 2004) has been implemented in three different graph grammars:

1. The first graph grammar (**Handling_RC**) deals with the description at the meta-level of how to generate compound models by handling the RCs identified in our approach. The above-mentioned set of routings includes sequence, concurrency, waterfall, conflict, accumulation, parallelism and synchronisation. Figure 3 shows some of the rules of this graph grammar (the implementation is composed of seven rules). We also illustrate how this graph grammar is defined by constructing GG rules implementing the specific coupling templates used to address the above RCs. In Table 1 in Appendix, we present the rule-based construction of two types of RCs considered in our research (sequence and accumulation). The nodes created or maintained by a specific coupling rule could be easily visualised due to the associated request for their attributes' values in the RHS.
2. The second graph grammar (**Standardization_CM**) converts the compound models generated as intermediary results in an iterative application of the Coupling step in the procedure proposed by the authors, into an aggregate model, matching the structure of the primitive model addressed in the above approach. The implementation of this grammar is made of 13 rules. The way to proceed is first to provide a *representative system of the input/output transitions set* of the resulting coupled models to be used by a subsequent iteration in applying the coupling

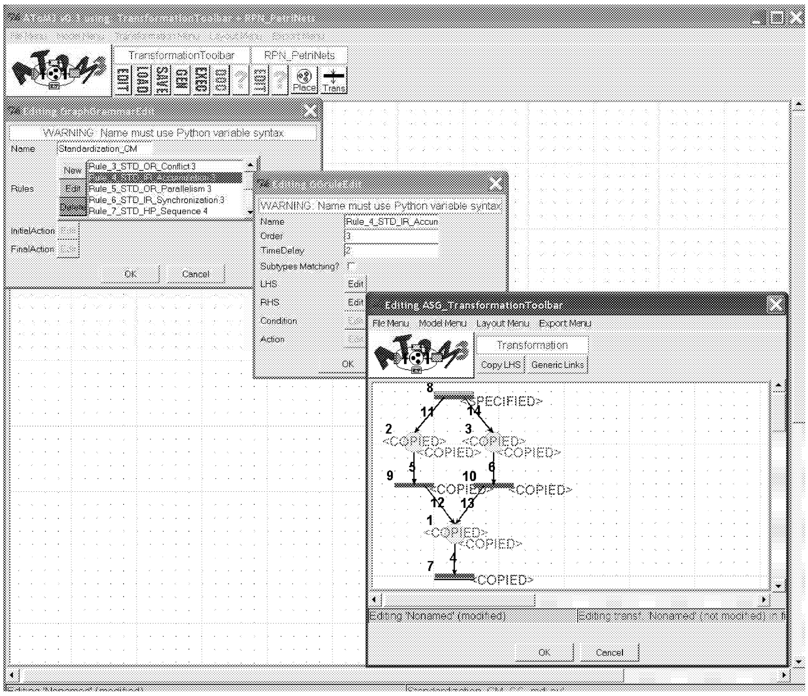


Name: Handling_RC
Rules: New / Edit / Delete
Rule 3: Waterfall 2
Rule 4: Conflict 2
Rule 5: Accumulation 2
Rule 6: Parallelism 2
Rule 7: Synchronization 2
InitialAction
FinalAction

Name: Rule 7: Synchronization
Order: 2
TimeDelay: 2
Subtypes Matching? □
LHS: Edit
RHS: Edit
Condition: Edit
Action: Edit
OK
Cancel

Editing 'Nonamed' (not modified)

Figure 3. Graph Grammar for Handling Routing Constructs



WARNING: Name must use Python variable syntax
Name: Standardization_CM
Rules: Rule_3_STD_OR_Conflict 3, Rule_5_STD_OR_Parallelism 3, Rule_6_STD_IR_Synchronization 3, Rule_7_STD_HP_Sequence 4

WARNING: Name must use Python variable syntax
Name: Rule_4_STD_IR_Accun
Order: 3
TimeDelay: 2
Subtypes Matching? □

Editing 'Nonamed' (modified)
Editing transf. 'Nonamed' (not modified)

Figure 4. Graph Grammar for Compound Models Standardization

mechanism. Other rules of this graph grammar deal with a simplified representation of the internal structure of the resulted model by using a hyper-place with "memory" (to trace all the hidden transitions and their associated input/output places). In Table 2 in Appendix, we illustrate how the aggregation process applies for two types of compound models generated by handling two RCs: sequence and accumulation.

3. The third graph grammar reverts to the detailed description of the resulted coupled model when the overall system description is achieved. By exploring the "memory" of the "standardization process" required by the resulted coupled models at different iterations in applying the Coupling step of the proposed construction, the hyper-places are substituted by the corresponding hidden transitions and their neighbourhood. Finally, the original input/output transitions sets are retrieved, by exploring their association with a given representative input/output transition set.

The final goal of our developments is to make the modelling process of a complex system automated from the appropriate graph grammars. At a certain extent this process will be made invisible to the user, who could be only involved in specifying the desired sequence of model manipulations.

CONCLUSIONS

In this paper the combination of meta-modelling and graph transformation to support visual modelling techniques is pleaded for. Special emphasis is put on the benefits of using the information in this meta-layer of modelling in the authors' current developments of a customised tool supporting the synthesis of compound PN models.

The ultimate goal of the developments addressed in this paper is to support the automated generation of multicomponent PN-based models. The advantages of using an automated tool to construct complex models in the PN domain according to the authors' method is clear: instead of building a complex PN-model from scratch, it is only necessary to identify the RCs addressed in the real system and specify the order they should be dealt with.

REFERENCES

AtoM^3 home page: http://atom3.cs.mcgill.ca/

Bobeanu, C.V. and Alla, H. 1998. "Modelling Basic Discrete Event Systems - A Key Factor in Supporting the Representation and Analysis of Complex Production Systems". In: Z. Binder, B.E. Hirsch and L.M. Aguilera (Eds.), Management and Control of Production and Logistics. Proceedings of the IFAC /IFIP Conference MCPL'97, vol. 1, pp. 193-198, Campinas -SP- Brazil, 1997. Oxford: Pergamon Publ.

Bobeanu, C.V., Kerckhoffs, E.J.H. and Van Landeghem, H. 2004. "Formal Modelling of Discrete Event Systems: A Holistic and Incremental Approach using Petri Nets". In*: ACM Transactions on Modeling and Computer Simulation*, Vol. 14, Issue 4 (October 2004), pp. 389-423.

Bobeanu, C.V. and Van Landeghem, H. 2005. "A Custom Tool Supporting Structural Modelling in the Petri Nets Domain". In: Feliz-Teixeira and Carvalho Brito (Eds.) *Proceedings of the 2005 European Simulation and Modelling Conference, ESMc2005,* pp. 579-583, Porto, Portugal, EUROSIS-ETI Publ.

Ehrig, H. and Padberg, J. 2004. "Graph Grammars and Petri Net Transformations". In: Desel, J., Reisig, W. and Rozenberg, G. (Eds.). *Lectures in Concurrency and Petri Nets. Advances in Petri Nets. Lecture Notes in Computer Science*, 3098, pp. 496-536, Springer.

Lara, J.de, Vangheluwe, H. 2004. "Defining visual notations and their manipulation through meta-modelling and graph transformation". In: Journal of Visual Languages & Computing, 15(2004), pp. 309-330.

Vangheluwe, H., Kerckhoffs, E.J.H. and Vansteenkiste, G. 2001. "Computer Automated Modelling of Complex Systems". In: Kerckhoffs, E.J.H. and Snorek, M. (Editors), *Proceedings of the 15th European Simulation Multiconference "Modelling and Simulation 2001", ESM'2001,* pp. 7-18, Prague, Czech Republic, June 6-9, SCS International Publ.

Van Landeghem, R. and Bobeanu, C.V. 2002. "Formal Modelling of Supply Chain: An Incremental Approach using Petri Nets". In: Verbraeck, A. and Krug, W. (Eds.). *Proceedings of the 14th European Simulation Symposium "Simulation in Industry",* pp 323-327, Dresden, Oct. 23-26, SCS International Publ.

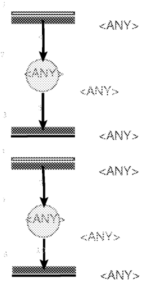
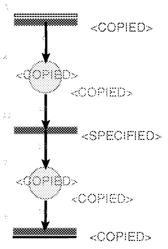
Van Landeghem, H. and Bobeanu, C.V. 2003. "Using Meta-modelling to Process Petri Nets Models of Supply Chains". In: Di Martino, B., Yang, L.T. and Bobeanu, C.V. (Eds.), *Proceedings of the 2003 European Simulation and Modelling Conference, ESM2003,* pp. 513-518, Naples, Italy, EUROSIS-ETI Publ.

Zeigler, B. P., Praehofer, H. and T. G. Kim. 2000. *Theory of Modeling and Simulation*, 2nd Edition, Academic Pres.

Appendix

Table 1. Handling_RC Graph Grammar: Coupled Models of Sequence and Accumulation

LHS	RHS	ACTION

Rule_1_Sequence

 	
	
	
		node = self.getMatched(graphID, self.RHS.nodeWithLabel(11))
		node.IsFusion = 1
		node.Trace = [self.getMatched(graphID, self.LHS.nodeWithLabel(3)), self.getMatched(graphID, self.LHS.nodeWithLabel(6))]

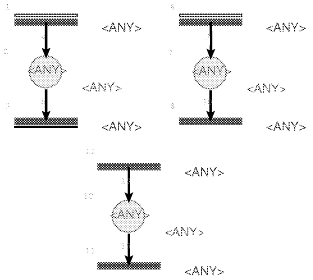
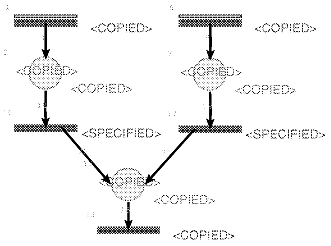
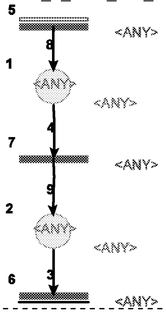
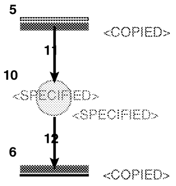
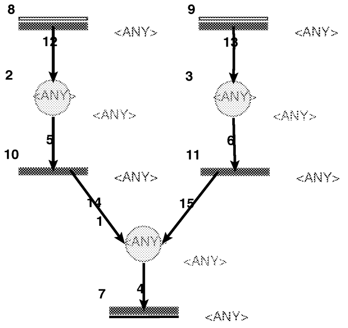
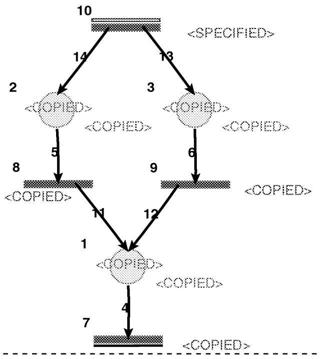
Rule_5_Accummulation

 	
	
	
		node = self.getMatched(graphID, self.RHS.nodeWithLabel(16))
		node.IsFusion = 1
		node.Trace = [self.getMatched(graphID, self.LHS.nodeWithLabel(3)), self.getMatched(graphID, self.LHS.nodeWithLabel(11))]
		node = self.getMatched(graphID, self.RHS.nodeWithLabel(17))
		node.IsFusion = 1
		node.Trace = [self.getMatched(graphID, self.LHS.nodeWithLabel(8)), self.getMatched(graphID, self.LHS.nodeWithLabel(11))]

Table 2. Standardization_CM Graph Grammar: Aggregate Representation of CM Sequence and Accumulation

LHS	RHS	ACTION

Rule_7_STD_HP_Sequence

 	
	
	
		node = self.getMatched(graphID, self.RHS.nodeWithLabel(10))
		node.refinementMap = [(self.getMatched(graphID, self.LHS.nodeWithLabel(7)), self.getMatched(graphID, self.LHS.nodeWithLabel(1)), self.getMatched(graphID, self.LHS.nodeWithLabel(2)))]

Rule_4_STD_IR_Accumulation

 8
9
<ANY>
<ANY>
12
13
2
3
<ANY>
<ANY>
<ANY>
<ANY>
5
6
10
11
<ANY>
<ANY>
14
15
1
<ANY>
<ANY>
7
4
<ANY>
	
	 10
	<SPECIFIED>
	14
	13
	2
	3
	<COPIED>
	<COPIED>
	<COPIED>
	<COPIED>
	5
	6
	8
	9
	<COPIED>
	<COPIED>
	11
	12
	1
	<COPIED>
	<COPIED>
	4
	7
	<COPIED>
		node = self.getMatched(graphID, self.RHS.nodeWithLabel(10))
		node.IsInput = 1
		node.setExtension = [self.getMatched(graphID, self.LHS.nodeWithLabel(8)), self.getMatched(graphID, self.LHS.nodeWithLabel(9))]

AUTHOR LISTING

AUTHOR LISTING

AUTHOR LISTING

AUTHOR LISTING