SCIENTIFIC PROGRAMME
KEYNOTE ADDRESS
DIDACTIC ROLE OF COMPUTER SIMULATION IN EDUCATION AND INDUSTRY

K. Heinz Weigl
Vienna University of Technology
A-1040 Vienna, Austria

E-mail: h.weigl@esla.at

ABSTRACT

The rapid change in today’s business world, the increasing numbers of complex systems, pressured managers forced to react faster while avoiding risk are not dramatically expanding the usage of simulation as desired by the professionals.

Simulation is today one of the most powerful analysis tools, to describe and analyze the behavior of a system, ask what-if questions about the real system and offers many and well recognized benefits. Although its use has grown over the almost past 50 years, major investments still are being made in new or modified systems without benefitting from the predictive power of simulation technology. We all not knowing whether those decision makers could save time, money and jobs in case of the usage of simulation. We all-specialists and professionals know, that it is inconceivable that any new designs or significant modifications of systems are built without the benefit of simulation. But- we were not able until now to bridge the gap between the simulation specialists and the many pragmatic, nontechnical analysts who need to understand and use this technology to solve real problems faster, cheaper and better.

Independent of the domain of interest, I am focusing on the transition of simulation from a highly specialized technology to one, like static flowcharts or spreadsheets, that is widely used by analysts and the majority of managers, not taken into account randomness as a critical element of a system, which is often more important than system design parameters.

Understanding the impact of randomness and variability in almost all business environments is extremely valuable in order to predict the future of systems performance more accurate.

Why is simulation not yet a key tool of the business strategy toolbox of large and small companies and their managers? The answer to that is the lack of education, time and resources and unawareness of the value that simulation adds to their business. Simulation is still not recognized in a wide range of industry and has not becoming commonplace as a standardized methodology throughout the industry.

One of the major reasons for that I detected in the curriculums of universities, where Simulation is not a required subject like mathematics or physics.

In Europe simulation slowly finds its way into the classroom within industrial engineering, logistics operations management etc. In the United States, simulation is becoming a required part of students’ curricula at many universities in departments such as industrial engineering.

In Europe only simulation idealists are initial catalysts for teaching simulation in a wide range of subjects, like manufacturing, control systems, electronics, mechatronics etc. The responsible individuals for curriculums in university programs have not yet detected Simulation as a valuable tool for closing the gap between theory and practice. Students often do not make connections between mathematical details and techniques, and the actual operation of engineering systems. From the teaching viewpoint there is still the tendency in teaching engineering to emphasize the mathematical statement of a problem and the steps required obtaining mathematical solutions.

This leads students getting lost in mathematical details without understanding underlying concepts and the actual engineering problems. The use of computers and simulation tools in the learning and teaching process should help to overcome this lack of understanding. In addition simulation methodologies should help to improve students problem solving skills. Simulation will add value during the
educational process to the quality of understanding of real world processes, supports capturing engineering knowledge and helps gaining experience faster now and in future. Simulation is not a sophisticated technology, no problem is too simple or too complex for using simulation, avoiding oversized costly and inefficient systems.

We all - professionals, specialists, teachers are playing NOW an important role to move Computer Simulation into the strategic toolbox of managers of all levels and companies of all sizes.

AUTHOR BIOGRAPHY

K. HEINZ WEIGL was born in Vienna, received his degrees in electrical engineering, his Ph.D. (Dr.techn) from the Faculty of mechanical and Industrial Engineering of the Vienna University of Technology and his Dr. Ing. Habil. in Production Logistics from the University of Technology Dresden. Dr. Weigl worked for Siemens in the high voltage motor division in Germany. In 1986 he started in the field of computer simulation, served in 1991 to 1993 on the board of directors of the Society for Computer Simulation in San Diego, USA. From 1996 to present he is leading the significant development project on Simulation and scheduling at the Vienna International Airport. Dr. Weigl serves as lecturer at numerous academical institutions in Europe. His main interest is in huge discrete event industrial models linked to scheduling systems.
AOSE
AND
SIMULATION
MODELING OF SOCIAL SYSTEMS WITH INGENIAS

Juan Pavón, Candelaria Sansores, Jorge Gómez-Sanz
Universidad Complutense Madrid
Facultad de Informática, Ciudad Universitaria s/n
Madrid 28040, Spain
E-mail: japvon@sip.ucm.es, csansores@fdi.ucm.es, jjgomez@sip.ucm.es

KEYWORDS
Agent based simulation, Agent based modeling, social simulation, agent development tools, model transformation, model driven development, INGENIAS Development Kit.

ABSTRACT

Most agent based simulation toolkits are based on the Java programming language and this makes their use difficult for social scientists, who are not usually skilled in computer programming. However, agent modeling concepts are not far from those which could be used for social modeling. This assumption is considered to propose the use of an agent-oriented language for the specification of social simulation models, and transforming (automatically) these models to code for some agent-based simulation toolkit. This is made with the INGENIAS Development Kit (IDK), a set of tools that provides a customizable model editor and modules for automatic code generation.

INTRODUCTION

There are good toolkits for agent based simulation (ABS). However, for many potential users of ABS, such as sociologists or economists, it is still very difficult to specify models with such tools. This is mainly because the task requires writing code in some, usually object-oriented, programming language. There are some attempts to facilitate the specification with graphical tools and a set of predefined behaviours. For instance, Sesam (www.simsesam.de/) provides a graphical interface to specify agent behaviours as a kind of state machines. And some rapid application development environments, for instance using the Python language in RePast Py (repast.sourceforge.net/repastpy/). But these are only applicable to simple models. Another issue is that the agent models initially supported by the different toolkits are rather simple, mostly reactive agents, whose behaviour is specified programatically. An exception to this is SMDL (sdml.cifm.org), which provides a declarative modeling language (although the learning curve seems to be not short).

On the other hand, there are many proposals of rich agent modeling languages in the agent-oriented software engineering domain. These languages are usually higher level, based on some graphical notation, and, in some cases, easily customizable. Their capabilities make them more suitable for depicting models than programming languages. Also, the supported agent model is richer, both at micro (agent) and macro (organization) levels, than in the agent based simulation toolkits. Furthermore, agent modeling concepts are closer to the abstractions that a sociologist or economist could use to model social systems. For these reasons, we consider that these languages should be considered for modeling social systems rather than plain program code.

If we use such kind of modeling languages, in order to perform the simulation we need either to execute the specification or transform it to some powerful simulation toolkit that could perform it. The execution means animating the specification, and being able to observe the execution to get some information on how the system evolves over time. This is not normally supported by agent-oriented software engineering tools. But agent-based simulation toolkits do this very well. So we are considering the automatic generation of code for such platforms by applying transformations from the more abstract agent models. This idea of transformation is in line with model driven development (MDD) practices.

With this perspective, we have been working in the modeling of complex social systems using an agent-oriented development methodology, INGENIAS (Pavón et. al 2005). This methodology is supported by a set of tools, the INGENIAS Development Kit (IDK, available at ingenias.sourceforge.net), which facilitates the edition of models and the definition of transformations for automatic code generation. The foundation of INGENIAS is the definition of multi-agent system (MAS) meta-models, which can be customized to particular application domains. This is interesting if we need to extend the existing notation to cope with new issues that may be required for the simulation of a social system.

In this paper we present how INGENIAS modeling language can be used to specify social systems and how transformations can be built towards one of the most well-known agent based simulation toolkits, RePast (repast.sourceforge.net). In the next section we discuss about what elements and concepts should be considered in a language for modeling of social systems. This is followed by a section where the INGENIAS modeling language is reviewed to see how it supports the requirements for social systems modeling, and how it can be extended. Then, a section describes how to make the transformation from INGENIAS social system specifications to RePast models. These models can be used for simulations on the RePast framework. This has been already experimented for RePast
and Mason (Sansores and Pavón 2005), in order to see the consequences of using different agent based simulation toolkits. The conclusions summarizes some of these results and discusses the limitations and prospective of this approach.

MODELING SOCIAL SYSTEMS

Computational simulation of social phenomena implies building computer programs that model the evolution of social processes. This involves the modeling of individuals and groups, and the processes of social interaction. The characteristics of social phenomena, like emotions, social pressure, etc. are difficult to be approached; therefore they can be modeled at some reasonable level of abstraction just for specific purposes.

We are specifically interested in modeling human behavior and individual decision making in society. This requires the ability to represent social interactions which give rise to the emergence of sociality like cooperation, competitions, groups, organization, etc. Therefore, the level of abstraction of the language we will be using is the individual’s social action and mind (Castelfranchi, 1998).

The language for modeling social systems can be defined on ontological concepts of the micro-sociology category of the sociological perspectives on Society. Under this perspective, a human being is capable of conscious thought and self-awareness. Human action is not simply a reaction to external stimuli, but the result of the meanings, theories, motives and interpretations brought into a social situation by the individual.

Accordingly to this, we need to conceptualize an individual with mental states and not just like a behavioral entity. This implies that representations of beliefs and goals of its own are not enough and beliefs and goals of the minds of others are required as well. We agree with Castelfranchi when he states that an individual should be modeled like a goal-oriented agent whose actions are internally regulated by goals and whose goals, decisions, and plans are based on beliefs. Both goals and beliefs are cognitive representations that can be internally generated, manipulated, and subject to inferences and reasoning.

Another fundamental concept is that of social action. As we have mentioned before, an individual performs actions that, depending of the level of awareness, could be simple reactive behavior or social actions. For an action to be considered social it needs to be goal-directed to another entity in a common shared world and that should be an active, autonomous goal-oriented entity itself, both entities interacting and perceiving each other as their similar. In this sense, we are not considering and action directed to the environment as social, although in certain cases it could be. For our purpose, directed towards modeling human behavior, we propose to use a classification of forms of activity and interpersonal relations in Sociology to model social actions. Social actions form the basis of social interactions, which is a dynamic changing sequence of social actions between individuals or groups. Social interactions form the basis of social relation, a multitude of social interactions regulated by social norms, between two or more people, with each having a social position and performing a social role.

Table 1 summarizes this classification and illustrates the concepts required to define the elements of our language.

<table>
<thead>
<tr>
<th>Social Actions</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Communication</td>
<td>Aimed at giving beliefs to the addressee</td>
</tr>
<tr>
<td>Rational</td>
<td>Planned action, taken after considering costs and consequences</td>
</tr>
<tr>
<td>Instrumental</td>
<td>Action that is taken because it leads to a valued goal, but with no thought of its consequences and often without consideration of the appropriateness of the means chosen to achieve it</td>
</tr>
<tr>
<td>Emotional</td>
<td>Action that is taken due to one’s emotions, to express personal feelings</td>
</tr>
<tr>
<td>Traditional</td>
<td>Action that is carried out due to traditions, because they are always carried out in such situation</td>
</tr>
</tbody>
</table>

With these concepts we will be able to define a language composed of individual types and their attributes as well as the social actions, interactions and relations between individuals and groups. The different instances of an individual type will give rise different individuals in a society with different roles, i.e. father of a family, mother, child, etc. It will also be possible to define groups, like families, as aggregates of the individual type.

The individual attributes may include emotional or character attributes. For example, a numeric stress level variable that determines the agent’s behavior in a given time. There are many possibilities of attributes to define an individual, and depending of their values the individual could perform a variety of predefined actions associated to these values.

Society and the environment it imposes, will include resources which the individual may count on, also the individual may have a set of actions to perform on the environment, like the action of using a resource for example.

INGENIAS MODELING LANGUAGE

INGENIAS is a methodology for the development of multi-agent systems (MAS) that integrates different results in the area. This integration is made by validation of methods and tools through experimentation in several applications during the last years. For this reason, INGENIAS assumes from the beginning the need to evolve the modelling language, the methods and the tools. And to facilitate this it bases on meta-modelling. The different tools of the INGENIAS Development Kit (IDK), are generated from MAS meta-model specifications. These are specifications of what elements the modelling language has and which constraints should apply in their use. With this framework, changes in
the meta-model specification are quickly reflected in the tools. This approach facilitates the evolution of INGENIAS and its adaptation to specific application domains, as it is the subject in this paper for social simulation.

The INGENIAS modelling language is structured in five packages, which represent the viewpoints from which a MAS can be regarded (see Figure 1): Organization, Agent, Goals-Tasks, Interactions, and Environment.

![Figure 1: MAS Viewpoints in INGENIAS](image)

The organization of a MAS establishes the framework where agents, resources, tasks and goals coexist. It defines structural relationships (groups, hierarchies), social norms (constraints and forms in the behaviour of agents and their interactions), and workflows (how agents collaborate when performing tasks in the organization).

Groups may contain agents, roles, resources or applications. There may be several ways to structure an organization. For instance, according to its functional needs. Or at the same time it can be structured by a geographical distribution. An agent, therefore, can belong to several groups at the same time. Assignment of elements to a group obeys some organizational purpose, i.e., because the grouping facilitates the definition of workflows or because its members have some common characteristics.

In general, the concept of role is used to provide more flexibility in the definition of organizations. A role represents functionality or services in an organization structure. Agents play roles in the organization. And several agents may play the same role, each one according to its abilities and strategies.

The functionality of the organization is defined by its purpose and tasks. An organization has one or more goals, and depends upon its agents to perform the necessary tasks to achieve them. How these tasks are related, and who is responsible of their execution, is defined in workflows. Workflows show the dynamics of the organization. They define associations among tasks and general information about their execution. For each task, a workflow defines which are its results, the agent or role responsible for its execution and which resources are required. This is useful to gain knowledge on the relationships between agents through tasks, and the assignment and availability of resources in an organization.

Both aspects, structural and dynamic, define the macro view of the MAS. This perspective facilitates the management of complex systems as it allows to determine the context and norms for the behaviour of agents, similarly to what happens in human organizations.

Agent behaviour is described in the agent viewpoint. It is determined by the agent mental state, a set of goals and beliefs. Also, an agent has a mental state processor, which allows the agent to decide which task to perform, and a mental state manager to create, modify and delete mental state entities. INGENIAS does not state specifically how to define the mental state processor as it considers that there may be many ways to implement it. For instance, it could be a rule based engine, a case based reasoning system, or a neural network. It depends on the needs of the application or the mechanism that best fits according to the developer.

Agents are intentional entities, this means that they act as they pursue some goals. As they are also social entities, they collaborate to satisfy organizational goals. When designing a MAS, it is possible to start with the identification of organization (system) goals. These goals can refine into simpler goals up to a level where it is possible to identify specific tasks that satisfy them. Another possibility is to identify individual goals for agents, which could be refined in a similar way. In both cases, there will be a relationship of goals and tasks, which is described in the goals-tasks viewpoint.

As social entities, agents interact. Their interactions can be produced in several ways, being the most common message passing, which is normally asynchronous, and shared spaces, where agents can act (produce modifications) and perceive (the modifications) as it is the case in shared tuple spaces. This is described in the interaction viewpoint. In INGENIAS, apart of indicating the types of messages and protocols in an interaction, what is important is to show the intentionality of the interaction: which goals are pursued by the parts in the interaction, and how this contributes to their satisfaction.

Finally, the environment is where agents perceive and act. Depending on the application, perception and acting can have very different meanings. The environment consists of a set of resources, applications, and other agents. In many situations, the environment can be specified as a set of application programming interfaces, which would be the classes that wrap it to allow interaction with it.

For social simulation, agents usually require to consider their location in the environment and the evolution of time. These two aspects have required a refinement of the environment viewpoint in INGENIAS.

The temporal perspective deals with the progress of time in the model when executing the simulation. In this case we are assuming that simulations are time driven rather than event driven as most agent based simulation toolkits work with this schema (a reason for this is that an event driven schema would require a central coordinator for events or a complex synchronization among agents). This means that
there is a need to model constant time steps to simulate the cycle perception-reaction of agents along the time.

The spatial perspective describes how agents are situated in the environment. In general, simulation toolkits provide two and three dimensional spaces with diverse configurations.

These extensions have required modifications in the original INGENIAS MAS meta-model and re-generation of the IDK tools. In this way, it has been possible to get a new personalized IDK for agent based simulation.

FROM INGENIAS TO REPAST

Repast (Recursive Porous Agent Simulation Toolkit) originally developed by (Collier et. al 2003) is a free open source programming framework created by Social Science Research Computing at the University of Chicago for the development of agent based simulation using Java language. It provides a class library for creating, running, displaying, and collecting data from the simulations. A simulation usually consists of a collection of agents of any type and a model that sets up and controls the execution of these agents’ behaviors according to a schedule. The execution of the simulation is divided into time steps or "ticks" in which the agents may play an action.

The specification of the agents and the environment in Repast is carried out by a set of variables and methods according to the object-oriented paradigm. It does not have intelligent agent concepts nor agent architectures, so an agent is an object without any restriction in relation to its internal architecture. However, it does provide specific libraries to implement functionalities like neural networks or genetic algorithms. Also, it provides facilities for easily developing graphical user interfaces, execution, monitoring, and presentation of results. The monitoring facilities are capable to visualize and to modify agent’s internal state dynamically as well as model properties at run time. The execution and presentation facilities led to visualize an animated representation of the simulation with a range of two-dimension agent environments and visualizations, provides tools to record snapshots of the display and social networks support tools as well, also it offers built-in simulation results logging and graphing tools.

Social abstractions like social actors, social actions, organizations, groups, etc, are missing in Repast. These abstractions are relevant for our purpose since we intend to model social systems in a high level modeling language and map these models into Repast computational models, accomplishing in this way the code generation task. In a model driven development approach, if modeling concepts in the platform independent models do not exist in the platform specific models, we have two options to solve this issue: one is to transform or adequate the high level language concepts or design concepts into existing platform specific concepts, or if they do not exist at all, like in the case of Repast toolkit, we have the second option which is to create them. If we do not create them in the specific implementation platform then we are going to use agents and social concepts for modeling social systems but not for implementing them. A negative side-effect is that there is absolutely no guarantee that what is being designed corresponds to what is being implemented and that the computational model is not leveraging the MAS paradigm. For this reason, we include a set of social and MAS concepts like a computational INGENIAS-SIM model built on top of Repast toolkit facilities, as we will see further on.

In the following, we describe how the generic transformation process of specifications with INGENIAS is performed, then the correspondence of INGENIAS elements with Repast elements.

The IDK supports transformation of specifications with IDK modules. Modules (or plugins) in INGENIAS are programs that process specifications and produce some output:

- **Source code**: There is an infrastructure that facilitates the transformation of the specification into source code. This is based on the definition of code templates for each target platform and information extraction procedures from the current models.
- **Reports**: Specifications can also be analysed to check, for instance, whether they have certain properties or whether special semantics (defined by the developer) are being respected, or to collect statistics of usage of different elements.
- **Modifications on current specification diagrams.** Though this feature is in beta stage, a module could insert and/or modify entities in the diagrams, or insert/delete diagrams. This feature is useful to define personal assistants that interact with the tool.
- **Other models.** As we produce source code, there is no difference in producing as well other models following different meta-models.

The mapping from INGENIAS models to Repast is implemented therefore by an IDK module. The module has to traverse specifications, extract information from specifications, and put the extracted information into templates of RePast agents. Environment constraints (such as spatial and temporal aspects) can also map to the main program.

The IDK module for Repast code generation is developed as an iterative process through several steps. This process can be applied similarly for other agent based simulation toolkits. The basis for code generation is the availability of code templates for the target platform, Repast in this case. This is usually the most difficult to obtain as it requires a good knowledge of how to implement agents in the target platform. Our experience has shown that this can be accomplished through an iterative process, in which the developer defines progressively the architecture of the code for the target platform and the transformations from specification to code templates. This process could be sketched in several steps, which have been applied to develop the module for Repast but could be used for modules that generate code for other platforms:

1. **Small initial prototype.** The process starts with a simple prototype of the simulation model. Initially,
the developer would centre into one or more features of the specification, easy to implement if possible. For instance, how to make an agent process an event. As a result, the developer gains knowledge on the target platform and has a prototype of an application on the target platform that realizes a small part of the specification with a selected set of features.

2. **Marking up the prototype code.** Looking at both the prototype and the specification, it is possible to identify parts of the prototype that match parts of the specification. As a result, the developer identifies possible mappings from the specification to the prototype code. This is reflected in a prototype code marked up with tags. The marked-up pieces of source code are called templates.

3. **Generating/modifying a module.** A module has to traverse the specification in order to obtain the information required to instantiate and fill in the prototype templates. The IDK provides an API for traversing specifications and Java packages for building modules. In concrete, the module engineer has to extend the class BasicCodeGeneratorImpl for the code generation module. Other support classes may be created as well.

4. **Deploying the module.** The resulting Java classes and templates of the module are put together into a jar file. This jar file is deployed in a specific folder where the IDK Editor can load it dynamically.

5. **Testing the module.** Testing is performed from the IDK Editor. By executing the module over the specification, the developer can check if the diagram is traversed properly and if all templates have been filled in as they should. Also, as templates demand concrete information, it may be possible that this is not present or that it is not expressed as it should. Therefore, it may turn out that the specification was not correct or incomplete. In this sense, any module can be useful to validate the specification against some completeness criteria. As a result, several kinds of problems may appear: with the code generated by the module, with the traversal of the specification, or with the specification itself.

6. **Debugging.** If something goes wrong, debug the prototype and go to:
   a. Step 2. If there is new code that was not marked up before.
   b. Step 3. If the failure was in the module and the data traversal.
   c. Step 4. If there was a failure in the prototype and could be solved without marking up the code again.

7. **Refinement and extension.** When the module is finished, it can translate diagram specifications into code or perform verification of some properties. However, the module performs these tasks with a reduced set of the diagram specification. The next step would be to take the code generated by the module and extend it so that it can satisfy other parts of the specification. Therefore, we would go back to step 1.

---

![Diagram of code generation process](image)

**Figure 2:** A simplified example of code generation

---

In this way, modules produce code using a template based approach. For RePast, there are several templates:

- A RePast agent template. This template implements a prototype of an INGENIAS agent which implements a drawable Repast interface for being displayed in a simulation and a RunnableAgent interface.
- A model template. This template implements the main class which sets up the simulation components, like the collection of agents and the scheduler. For complicated scheduling mechanisms we could have a separately scheduler template.
• An environment template. This template implements the spaceless environment for agents without physics references as well as Repast space environment where spatial agents interact.
• A Role, Goal and Task templates which are usually embedded in the agent template.

The way the module works is shown in Figure 2. A developer defines a template of a RePast agent (steps 1 and 2) and extracts data from the MAS specification (step 3) to generate the code. A RePast agent needs a goal, a role, and a name. As a result, we get a Java Repast agent, which is instantiated from the Repast agent template.

These elements are configured within a module and deployed in the IDK (step 4), and tested over the specification of a MAS (step 5). As a result of the testing, we would obtain the generated code presented in Figure 2.

As we have said before, Repast lacks many concepts necessary to model social actors, although it gives users complete flexibility as to how they specify the properties and behaviors of agents. This flexibility allows Repast to be applied to a wide range of domains. However, it also means the burden of agent design and implementation is passed on to the simulation developer. While this may not be much of a problem where simple agents will suffice, in many applications it is useful to provide agents with more sophisticated capabilities; goal-directed agents that adapt to changing circumstances are desirable in many contexts.

The following figures depict: 1) the Java module developed to provide goal-directed behavior specifically for agents in Repast Simulations, this is, the computational INGENIAS-SIM model, 2) the main Repast elements and their INGENIAS specification elements. The Java module is the basis of Repast templates and relaxes the knowledge required to build agent-based simulations with Repast.

A typical simulation written with Repast will have at least three classes, the agent class, the model class and the space class. The Repast agent class will be largely simulation specific. If the agent is to be displayed, it should implement one of the Drawable interfaces. The Repast model class sets up and controls both the representational (variables like initial parameters) and infrastructure (variables such as a schedule) parts of a Repast simulation. All Repast simulation models must implement the SimModel interface. Repast provides an abstract class SimModelImpl that partially implements this interface and it is expected that most if not all models will extend this class. Figure 3 shows these Repast classes in package uchicago.src.sim which represent the main elements to be mapped by INGENIAS-SIM elements.

Also Figure 3 illustrates INGENIAS-SIM elements in package ingenias.sim.model, in this package the model class contains all the agents of a simulation and stores them in a list, referencing them via the RunnableAgent interface, it implements SimModel interface required methods as well as other methods that divide the infrastructure and representation setup process into coherent groups. It also provides features for creating and setting up the components of the simulation, like agents and model’s parameters. The RunnableAgent interface provides a standard access point for referencing agents by the model, the run() method of all agents in a simulation is called in each simulation step. The Agent class represents the social actors in a simulation, besides implementing the Repast Drawable interface it must conforms the RunnableAgent interface implementing the run() method, where agents sensing, deliberating and acting cycle activities are called. The agent class also includes sophisticated features like goal-dynamics, roles, actions, tasks and interactions not showed in this figure but included in the computational INGENIAS-SIM model.

Figure 3: Repast-INGENIAS-SIM model - agent elements

The Repast class space controls the environment in which the action takes place, in Repast it is theoretically possible to omit the explicit Space object, but this is rarely done. This means that agents are always immersed in a space relationship relative to each other, represented with one of the space classes of the package uchicago.src.sim depicted in Figure 4, which also shows the corresponding INGENIAS-SIM elements.

We provide a broader view of the environment. Instead of stating that the agents are physically in the environment we also give the possibility to define spaceless agents. In this sense, the environment will be notional more than physical. The space environments we support are more flexible with
respect to different representations, achieving this flexibility through an `EnvironmentInterface` interface, which offers a standardized access point for all environments implementations. As illustrated in Figure 4, the package `ingenias.sim.model` contains this interface and a concrete `Environment` class which implements it; the environment class uses a space Repast object `Object2DGrid` to implement the interface, this object represent a two-dimension grid space although it can be easily replaced by any other Repast spaces. The spaceless environments are supported with the aid of an environment class not depicted in Figure 4 but which provides methods for manipulating the environment by the agents, for example to locate other agents, to use a resource, etc.

![Figure 4: Repast-INGENIAS-SIM environment elements](image)

Within a Repast simulation all state changes are the responsibility of a `Scheduling` mechanism implemented with the `Schedule` class. The mechanism consists of setting up methods calls on objects to occur at certain times. These method calls must be wrapped by sub-classes of the `BasicAction` class. A `BasicAction` consist of some variables used by the scheduler and an abstract `execute()` method. Any classes that sub-class a `BasicAction` must implement this method. It is in this method that the actual method call or calls to be scheduled should occur. These two Repast classes represent the main elements of Repast scheduling and can be observed in Figure 5 in package `uchicago.src.sim.engine`.

As explained in (Sansores and Pavón 2005) this scheduling mechanism is contrary to some of the ideas that motivated a multi-agent based approach since creating a list of actions to be taken at each time step looks more like a central coordinator approach. For this reason, we do not schedule method calls on agents, rather we schedule the agents themselves, in `ingenias.sim.model` package illustrated in Figure 5 we can observe how we achieve this and the INGENIAS-SIM elements and their relationships with the corresponding Repast elements.

In `ingenias.sim.model` package, the `Model` class contains an inner class named `ModelRunner` which is itself a subclass of Repast `BasicAction` class. The `ModelRunner` can therefore be scheduled each time step and the implemented `execute()` method will be executed each step, is in this method where we call the `run()` method of each agent in the simulation, this is, each agent in `agentList` list. This means that is the agent’s internal structure where we have to include a decision mechanism of which actions to execute in each time step, in contrast to Repast which suggests calling actions in agents accordingly to a predefined listed actions in the model, of course our point of view although is better aligned with the MAS paradigm, it is certainly a more difficult way of timing agents’ actions. In the case of simulations where modelers need to have control over the artificial society to observe what is happening in every moment and what parameters influence the results, it is always possible to combine both scheduling approaches.

![Figure 5: Repast-INGENIAS-SIM scheduling elements](image)

Also, we can observe in Figure 5 the `Goal` class which are pursued by an agent and the `Role` class which may be played by the agent. The role also pursues goals. An individual agent can adapt its capabilities and observe specialization of roles based on a reinforcement mechanism, in this way changing dynamically roles played. The `Task` class allows defining a set of tasks to achieve the different Goals an agent.
is pursuing, thus an agent is related with a set of tasks to perform. A task may be composed of one or more actions defined with the Action class; the deliberating mechanism has to decide what action to execute depending of goal an agent is pursuing in a given time, an example of actions could be sending a message, moving in the environment, or application domain actions like moving an object for example.

CONCLUSIONS

The use of INGENIAS as shown in this paper facilitates the modelling of agent based social systems and its simulation on existing agent based simulation toolkits. This allows social scientists to avoid writing code in a programming language and describing models in a more intuitive language, closer to their problem domain.

Another advantage is the ability to replicate models on different simulation toolkits. As modelling is performed with a graphical language and then transformed into code, when this transformation is performed for several simulation environments it will be possible to contrast results. This has been already explored in other work (Sansores and Pavón 2005) with RePast and Mason, where we have studied the effects of different scheduling strategies in the simulation.

An interesting issue of INGENIAS to consider for further work is the ability to extend the INGENIAS MAS metamodel to create domain oriented specification languages. This would facilitate even further the modelling activity of social scientists as they could use concrete domain concepts rather than pure agent concepts from INGENIAS. This work has to be done with domain experts (sociologists) and this is a task that we are currently addressing.

ACKNOWLEDGEMENTS

This work has been developed with support of the Consejo Nacional de Ciencia y Tecnología (CONACYT) from México and the project TIN2005-08501-C03-01, funded by the Spanish Council for Science and Technology.

REFERENCES


BIOGRAPHY

The authors of this paper collaborate in the grasia! research group (http://grasia.fdi.ucm.es). They have a pragmatic view of agent technology, so they focus on software engineering issues (methods and tools), which are the result of their experience in different application domains (information recovery, intelligent interfaces, workflow management systems, resource planning and monitoring, ubiquitous computing, and social simulation). The main result of this work is the INGENIAS methodology for the development and deployment of multi-agent systems. INGENIAS provides a framework for research on several areas. Currently, focus is given to the study of social properties in multi-agent systems and simulation of complex systems.
SIMULATION-DRIVEN DEVELOPMENT OF MULTI-AGENT SYSTEMS

Giancarlo Fortino, Alfredo Garro, Wilma Russo
Dipartimento di Elettronica, Informatica e Sistemistica
Università della Calabria
Via P.Bucci, cubo 41C, 87036 Rende (CS)
Italy
E-mail: {g.fortino, garro, w.russo}@unical.it

Roberto Caico, Massimo Cossentino, Francesco Termine
ICAR - Italian National Research Council
Viale delle Scienze, 90128 Palermo
Italy
E-mail: cossentino@pa.icar.cnr.it, robertocaico@libero.it,
francescotermine@alice.it

KEYWORDS
Multi-Agent Systems, Discrete-Event Simulation, Agent Oriented Software Engineering

ABSTRACT
This paper presents a simulation-driven development process for multi-agent systems (MAS) which integrates a Statecharts-based simulation methodology into the well known and established PASSI methodology. It can be effectively used as an experimental tool in the context of the Agent Oriented Software Engineering for quantifying the benefits of using simulation for MAS development. To exemplify this process and demonstrate its effectiveness, a case study concerning with the design and simulation of a complex MAS is defined and detailed.

1. INTRODUCTION
Simulation is being greatly applied in many industrial fields, such as aerospace, automotive or energy production, but its application in support of software products and processes is to date still under considered. Despite of its limited exploitation in software engineering, Simulation has been recognized to be an effective tool to support software engineering experimentations involving requirements management, project management, training, process improvement, architecture and COTS (Commercial Off-The-Shelf) integration, product-line practices, risk management, and acquisition management (Christie 1999; Mayrhofer 1993). With the emergence of Agent Oriented Software Engineering (AOSE) as a new discipline (Luck et al. 2004) which aims at identifying and defining models and techniques suitable for the development of complex software systems in terms of MASs (Multi-Agent Systems), we wonder if Simulation could play a more strategic role in the development of MASs than that played in the development of traditional and/or conventional software systems, and, more specifically, if Simulation could provide a substantial added-value when applied to support the development process of MASs (Uhrmacher 2002).

The answer to our first question lies in the complexity of MASs with respect to the complexity of traditional software systems. A MAS is a system composed of several agents, capable of reaching goals that are difficult to achieve by an individual system (Woolridge 2002). MASs can manifest self-organization and complex behaviors even when the individual strategies of all their agents are simple. Thus, the use of Simulation can be crucial in the analysis of the MAS under-development at different scales of observation (macro, micro and meso levels) (Zambonelli and Omicini 2004) and, also, for the discovery of emergent properties which were not taken into account or were not considered at all in the design phase.

To answer the second question we need to quantify the claimed added-value in using simulation for MAS development through actual experimentations covering the whole software development lifecycle of MASs: requirements capture, analysis, design, implementation, deployment, and testing. To date a few of MAS development processes have been proposed in the literature (Electronic Institutions (Sierra et al. 2004), DynDEVS/James (Rohl and Uhrmacher 2004), CaseLP (Martelli et al. 1999), GAIA/MASSIMO (Fortino et al. 2005a), TuCSoN/Simulation (Gardelli et al. 2005), Joint Measure (Sarjoughian et al. 2001), etc) which incorporate Simulation to support the design phase of the MAS development lifecycle with the main focus on the validation and performance evaluation of the designed MAS model. However, to quantify the benefits of using Simulation for MAS development further research work need to be carried out in the aforementioned direction and in further directions encompassing all the phases of the MAS development lifecycle. The major benefits would be product quality improvement and project risk minimization. These would derive from the use of Simulation in pinning down MAS requirements early in the development lifecycle, in testing out alternate modifications of requirements, in safely examining alternate architectures and designs, and in gaining insights with timing, resource usage and bottlenecks.

In this paper we propose a simulation-driven development process which is obtained by integrating a Statecharts-based simulation methodology for MASs (Fortino et al. 2005a, Fortino et al. 2005b) with PASSI, a well-known development process for MASs (Cossentino 2005). This allows us, on one hand, to enrich PASSI with the potential benefits deriving from the exploitation of Simulation and, from another hand, to concretely experiment with a process supporting simulation-driven development of MAS. The obtained development process is exemplified through a case study concerning with the design and simulation of a MAS which represents a consumer-driven e-marketplace (CEM). In particular, the simulation phase allows for the validation of the correct behavior of the CEM under-development and for the evaluation of the CEM efficiency, in terms of completion time for buying a product, and efficacy, in terms of probability of buying a product at the desired price.

The rest of the paper is organized as follows. Section 2 describes the proposed agent-oriented simulation-driven development process. Section 3 is devoted to detail the proposed case study. Section 4 discusses some related agent-based design and development approaches incorporating simulation. Finally conclusions are drawn and directions of future research briefly elucidated.
2. AN AGENT-ORIENTED SIMULATION-DRIVEN DEVELOPMENT PROCESS

In this section we present an agent-oriented simulation-driven development process for building MAS. It is obtained through the integration of MASSIMO (Multi-Agent System SIulator framewOrk) and its supporting Statecharts-based simulation methodology (Fortino et al. 2005a) into PASSI (Process for Agent Societies Specification and Implementation), a development process for MAS (Cossentino 2005). The so obtained development process can use simulation to support the following phases of PASSI: system requirements, agent society and agent implementation. In particular, in this paper, we concentrate on the simulation of the Agent Implementation Model, i.e. the work product of the agent implementation phase describing the complex structure and behaviour of the MAS under-development, both for validation and for performance evaluation purposes. In the following subsections we first provide a brief description of PASSI and the Statecharts-based simulation methodology, and, then, we present their integration.

2.1. PASSI (Process for Agent Societies Specification and Implementation)

The PASSI methodology is a step-by-step requirements-to-code methodology for designing and developing multi-agent societies. It adopts design models and concepts from the UML that is adapted in order to represent the different elements and abstractions of a multi-agent system. The methodology is supported by the PASSI Toolkit (PTK), a Rational Rose plug-in, and by a repository of agent patterns. In PASSI, during the initial steps of the design, an agent is seen as an autonomous entity capable of pursuing an objective through its autonomous decisions, actions and social relationships. This helps in preparing a solution that is later implemented referring to the agent as a significant software unit. An agent may undertake several functional roles during interactions with other agents to achieve its goals. A role is a collection of tasks performed by the agent in pursuing a sub-goal or offering some service to the other members of the society. A task is defined as a purposeful unit of individual or interactive behavior. Each agent has a representation of the world in terms of an ontology that is also referred to in all the messages the agents exchange. PASSI is composed of the following five models regarding the different abstraction levels of the process (see Fig. 1):

- **System Requirements Model.** The initial part of this model is similar to other common object-oriented methodologies (requirements analysis phase), then an agent-based solution to the problem is drafted by assigning system functionalities to agents.
- **Agent Society Model.** This describes the details of the system solution in terms of agent society concepts like ontology, communications and roles.
- **Agent Implementation Model.** The previous models are used to obtain a detailed description of the agent society in terms of both structure and behavior that can be used to produce the code of the system.

- **Code Model.** In order to streamline and speed up the development of a new system, code is partially obtained from the application of patterns. A conventional code completion activity is then carried out.

- **Deployment Model.** Mobile agents require that a specific attention is paid to the specification of their needs in terms of both software environments (e.g., libraries available in the host platform), hardware capabilities and performance (e.g., amount of available network bandwidth); these are the issues defined in the deployment model.

![Image of the different steps and models of PASSI](image1)

Figure 1: The different steps and models of PASSI

2.2. A Statecharts-based simulation methodology for multi-agent systems

The simulation methodology (Fortino et al. 2005a) is based on the following three iterable phases: **Modeling**, **Coding** and **Simulation** of the MAS under-development (see Figure 2).

![Image of the simulation process](image2)

Figure 2: The simulation process

The **Modeling** phase is enabled by the Distilled StateCharts (DSCs) formalism (Fortino et al. 2004) which supports the specification of the behavior of the agent types and the interaction protocols among the agent types of a MAS. DSCs were derived from Statecharts (Harel and Gery 1997) and allow for the specification of the behavior of event-driven lightweight agents (ELAs) which are single-threaded entities capable of transparent migration and executing chains of atomic actions.

The DSC-based specification of a MAS, denoted as $\text{MAS}_{\text{DSC}}$, is expressed as $\text{MAS}_{\text{DSC}} = \{\text{Beh}(\text{AT}_i), \ldots, \text{Beh}(\text{AT}_n)\}$, where $\text{Beh}(\text{AT}_i) = \langle \text{S}^{\text{beh}}(\text{AT}_i), \text{E}^{\text{beh}}(\text{AT}_i)\rangle$ is the DSC specification of the dynamic behavior of the $i$-th agent type. In particular, $\text{S}^{\text{beh}}(\text{AT}_i)$ is a hierarchical state machine incorporating the activity and the event handling of the $i$-th agent type and $\text{E}^{\text{beh}}(\text{AT}_i)$ is the related set of events to be handled triggering state transitions in $\text{S}^{\text{beh}}(\text{AT}_i)$. In particular, $\text{S}^{\text{beh}}(\text{AT}_i)$ is designed on the basis of a template compliant with the FIPA agent lifecycle (FIPA 2001) (see Figure 3). The Active Distilled StateChart (ADSC), inside the Active state, is to be refined by the agent designer. The deep history connector (H*) inside the Active state allows for agent migration based on a coarse-grained strong mobility model (Fortino et al 2004).

The **Coding** (or prototyping) phase is supported by the Mobile Active Object Framework (MAO Framework) (Fortino et al...
2.3. Integrating MASSIMO into PASSI

Although the simulation methodology overviewed in the previous section could be used to validate the work products of the system requirements, agent society and agent implementation phases of PASSI, it is currently used for the validation of the Agent Implementation Model (AIM). PASSI is therefore enhanced with a further “step” involving the simulation of the AIM which must previously be translated into a MASDSC.

The semi-automatic translation process of the AIM (see Fig. 1) into the MASDSC is carried out as follows:

- The ATs are derived from the agent types of the Multi-Agent Structure Definition (MASD) through a one-to-one mapping.
- The interactions in terms of events exchanged between the AT’s are derived from the Multi-Agent Behavior Definition (MABD).
- The Beh(AT) VI is derived from the SASD (Single-Agent Structure Definition) and the SABD (Single-Agent Behavior Definition) of the i-th agent type.

A translation example based on the proposed case study will be presented in section 3.2. After the simulation phase, the designers can either proceed with the remaining part of the PASSI process, if they want to implement the software final release, or use the results of the simulation to feedback the System Requirement phase and/or the Agent Society phase.

3. A CASE STUDY

This section shows the application of the proposed approach to the analysis and design of a Consumer-driven E-Marketplace (CEM) system. A CEM system is a distributed software system which provides e-commerce services to end-users (or consumers) which drive the exchange of goods within the e-Marketplace. In particular, users according to their needs, browse the e-Marketplace, search for the vendors offering a given product, evaluate the vendors’ offers, contract product price with the vendors and, finally, decide to buy a product from a selected vendor. The payment phase is supported by an e-cash-based system mediated by a bank.

In subsection 3.1 PASSI is used to design the CEM. Subsection 3.2 shows the translation of a single agent behaviour description into a DSC model. Finally subsection 3.3 shows the simulation phase for a given CEM scenario.

3.1 Designing the CEM system with PASSI

In the following subsections the obtained system requirements, agent society and agent implementation models will be described.

3.1.1 The System Requirements Model

The System Requirements Model is a model of the system requirements in terms of agency and purpose. The methodology is use case driven and starts with the requirements analysis, where the designer models the system as a set of use case diagrams. Some of these diagrams, the Domain (Requirements) Description diagrams, are drawn to represent the actors and the use cases identified for the system. A use case represents a portion of the system.
behavior while an actor is an external entity interacting with the system; we identified the actors User, Vendor and Bank. In PASSI, each agent receives the responsibility for a part of the functionalities of the whole system; this is represented in a use case diagram, called Agent Identification (AId) diagram, by grouping some of the use cases within a package and giving it the name of the agent.

Figure 4 depicts the AId diagram for our system which includes the following identified agents:
- **User Assistant Agent (UAA)** is associated with a user and assists her/him in looking for a specific product that meets her/his needs and buying the product according to a specific buying policy.
- **Yellow Pages Agent (YPA)** represents an entry point of the federated yellow pages service (or “Yellow Pages”) which provides the location of agents selling a given product.
- **Vendor Agent (VA)** represents the vendor of specific goods.
- **Mobile Consumer Agent (MCA)** is an autonomous mobile agent dealing with searching, contracting, evaluation, and payment of goods.
- **Access Point Agent (APA)** represents the entry point for the e-marketplace. It accepts requests for buying a product from a registered UAA and fulfills them by generating a specific MCA.
- **Bank Agent (BA)** represents a reference bank of MCA and VA.

In the reported diagram, these agents are displayed as packages containing the use cases coming from the Domain (Requirements) Description Diagram that has been omitted because of space concerns. Each agent is responsible for accomplishing the functionalities associated with the use cases included in its package. Because of the specific nature of this diagram (a functional view), we cannot describe here agents interactions; this consideration finds an exception in the communicate relationship substituting the include/extend relationships occurring between use cases of different agents (being an agent an autonomous entity, it makes no sense to design an include dependency between two different ones).

![Diagram of AId diagram](image)

**Figure 4:** The AId diagram for the proposed case study

Once all the use cases have been assigned to agents that will be responsible for accomplishing them, the designer can explore the scenarios in which these agents will be involved. We usually do it with a set of UML sequence diagrams (Role Identification diagrams); in these diagrams each agent may be involved in several different activities and may appear more than once in each scenario playing different roles.

![Diagram of UML sequence diagrams](image)

**Figure 5:** A portion of a RId diagram regarding a specific product vendors search scenario

An example of a Role Identification (RId) diagram is shown in Figure 5 where the APA, playing the role of UserRequestValidatorAndForwarder, after validating the order, forwards it to the MCA, playing the Searcher role; hence the MCA asks for the vendor list to the YPA. Once the MCA gets the list, it contacts all the VAs and asks them for their offers.

The initial description of the dynamic behavior of each agent is the last step of the System Requirements Model. This phase is performed with a set of Task Specification Diagrams (one for each identified agent). The Task Specification Diagram is a UML activity diagram that represents the agent activity plan using two swim-lanes (see Figure 6): the right-hand contains a collection of roles including activities, while the left-hand reports some roles from other agents involved in interactions with this one; in this diagram the activities performed by the agent within each of its roles (Searcher, Contr&Eval, Payer, Reporter) are hidden.

The example reported in Figure 6, regards the MCA that is involved in searching the vendors list through a query to the YPA (Searcher role), then in the contracting and evaluation phase (Contr&Eval role) with the VA, in buying the product from the best bidder (Payer role) and in reporting (Reporter role) the transaction results to the UAA. At this point the MCA can play again its first role or it can be terminated.

![Diagram of Task Specification diagram for the MCA](image)

**Figure 6:** The Task Specification diagram for the MCA

### 3.1.2 The Agent Society Model

The next PASSI model is the Agent Society Model that represents social interactions and dependencies among agents involved in the solution. It begins with the ontology design
that is performed in the Domain Ontology Description (DOD) phase with the use of a class diagram. A DOD diagram describes the ontology in terms of concepts (categories, entities of the domain), predicates (assertions on properties of concepts) and actions (performed in the domain). This diagram can also be regarded as an XML schema that can be used to obtain a Resource Description Framework (RDF) (FIPA 2001; RDF 1999) encoding of the ontological structure.

![Image of DOD diagram](image)

**Figure 7: A portion of the DOD diagram**

Figure 7 shows a portion of the DOD diagram obtained for the case study, where we can see some of the concepts, predicates and actions used to define the problem domain. For instance the Vendor concept (representing the vendor of the real-world scenario) is related with the Product(s) it sells. A vendor registers its products in the agent-based yellow pages service by executing the RegisterProduct action which is performed by the VA and its outcome received by the YPA.

The Communication Ontology Description (COD) is a class diagram that shows all agents and all of their communications (relationships among agents). This diagram is drawn starting from the results of the Ald phase. A class is introduced for each agent, and an association is introduced for each communication between two agents. Obviously, according to the principles of an iterative/incremental design process, in further refinement communications can be added, merged or removed as a consequence of the arising needs. Being communications a way to exchange knowledge, it is also important to introduce the proper data structure (coming from the entities described in the DOD) in each agent in order to store it. The association line that represents each communication is drawn from the initiator of the conversation to the other agent (participant) as can be deduced from the description of their interaction performed in the Rd phase. Each communication is characterized by three attributes, (Ontology, Agent Interaction Protocol and Content Language) which we group into an association class. This is the characterization of the communication itself and its name is used to uniquely identify it (this communication can have, obviously, several instances at runtime).

In Figure 8 an example of COD diagram is reported. It represents three agents (APA, VA, MCA) and two communications among them (Forward_Product_Request, Offer_Request). In particular, the Offer_Request communication happens when, in the scenario reported in Figure 5, the MCA asks the VA for the best offer.

![Image of COD diagram](image)

**Figure 8: A portion of the COD diagram**

This communication refers to the OfferPrice predicate from the ontology of Figure 7 and adopts the FIPAQuery agent interaction protocol and the RDF content language.

Roles played by agents during the interaction (as described in the Rd diagrams) are reported at the beginning and the end of the association line.

As it has already been discussed in previous sub-section, PASSI roles are initially identified in the Ald diagrams. Their definition is then completed with the Role Description (RD) diagram that is a UML class diagram in which classes are used to represent roles; each role uses several elementary tasks to implement its complex behavior; finally, roles are grouped in packages representing agents.

The Agent Society Model ends with the Protocol Description phase which is required only when the FIPA standard protocols are not sufficient to solve some communication problem (this is not the case for our case study).

### 3.1.2 The Agent Implementation Model

The **Agent Implementation Model** is a model of the solution architecture. It is composed of two different phases, each performed at both the multi- and single-agent level of abstraction. The multi-agent level deals with the agent society and reports low details about agent implementation; however, it fittingly documents the overall structure of the system (behaviors of each agent, communications, etc.).

The single-agent level of abstraction focuses on the implementation details of each agent and specifies whatever is needed in order to prepare the coding phase.

The two phases that are performed at the multi- and single-agent levels are:

- Agent Structure Definition (ASD); that uses conventional class diagrams to describe the structure of solution agent classes;
- Agent Behavior Description (ABD); that uses activity diagrams or state-charts to describe the behavior of individual agents.

In the **Multi-Agent Structure Definition** (MASD) diagram, automatically generated by the PTK tool on the basis of the previous diagrams, the focus is on the general architecture of the system. The MASD diagram is an overview of the multi-agent system from the structural point of view. In this diagram, agents are represented as classes with their behaviors in the operation compartment and attributes specifying the agent knowledge.
In Figure 9 we report the portion of the MASD describing the structure of the VA, MCA and APA agents. It is worth to note that the VA is in relationship with an (human) actor; this is an extension of UML that we consider useful to represent in a unique diagram all the agents relationships (communications and GUI-based interactions with the user).

The agent behavior at the multi-agent level is described by the Multi-Agent Behavior Description (MABD) diagram. This is a UML activity diagram used to illustrate the dynamics of the system during the agents’ lifecycle. In the diagram, the involved agents and their tasks are represented with swim-lanes, while operations are displayed as activities. In this diagram, transitions among activities represent events like method invocations (if relating activities in the same swim-lane), new behavior instantiations/invocations (if relating activities of the same agent but in different swim-lanes) or messages (if activities from two different agents are involved). Figure 10 reports a portion of the obtained MABD diagram which illustrates the activities occurring during the Request communication between MCA and YPA and the Query communication between MCA and VA. In particular, it describes the request of the vendors list from the MCA to the YPA; then, the MCA moves to the VA location and begins the contract phase by asking for an offer which is soon after evaluated.

Although this representation is very useful and gives a complete overview of the MAS, it is not sufficient to detail the algorithm implemented in each of the agents within the activities. This further refinement step is usually done through the Single-Agent Behavior Description, which is an activity diagram in which the swimlanes represent the tasks performed by the agent during its lifecycle. Figure 11 shows the general SABD of the MCA, which should be self-explanatory. This is the point where we should consider whether to implement the system or to simulate it. For simulation purposes, we have to translate the SABD of each agent into a DSC. In the next subsection we therefore use the DSC formalism to detail the SABD of a specific MCA.

3.2. The DSC-based SABD of an MCA

On the basis of the SABD shown in Figure 11, two types of DSC-based MCA have been implemented: (i) Itinerary Consumer Agent (ICA), which performs the Searching and Contr&Eval phases by sequentially moving from one location to another within the e-Marketplace; (ii) Parallel Consumer Agent (PCA), which performs the Searching and Contr&Eval phases by means of a set of parallel mobile agents called workers (Fortino et al. 2005a).

Figure 12 shows the ADSC (see section 2.2) of the ICA obtained from the SABD of the MCA (see Fig. 11) and from the MABD (see Fig. 10).
In the ADSC of the ICA the events, which can be internal (i.e. self-driving the agent behavior) or external (i.e. targeting another agent), are generated through the primitive `generate(⟨mevent⟩, ⟨param⟩)`, where `mevent` is an event instance and `param` is the list of formal parameters of `mevent`. In addition, events are asynchronously received and processed according to a run-to-completion semantics (i.e. an event can be processed only if the processing of the previous event has been fully completed) (Fortino et al. 2004).

The names of the composite states of the ADSC corresponds to the names of the tasks of the MCA shown in the related SABD. For the sake of modularity the SEARCHING and CONTR&EVAL states are embodied into the SEARCH&BUY state.

The activities reported in the SABD are implemented by the action chains of the ADSC; the association between activities and action chains is reported in Table 1.

<table>
<thead>
<tr>
<th>SABD ACTIVITY</th>
<th>ADSC ACTION</th>
<th>ACTION CHAIN</th>
</tr>
</thead>
<tbody>
<tr>
<td>MoveTo Next YPA Target</td>
<td>ac1, ac2</td>
<td></td>
</tr>
<tr>
<td>Request YPA List</td>
<td>ac3</td>
<td></td>
</tr>
<tr>
<td>Process YPA Reply</td>
<td>ac4</td>
<td></td>
</tr>
<tr>
<td>MoveTo Next VATarget</td>
<td>acl</td>
<td></td>
</tr>
<tr>
<td>Request An Offer From VATarget</td>
<td>ac5</td>
<td></td>
</tr>
<tr>
<td>Evaluate VATOffer</td>
<td>ac6</td>
<td></td>
</tr>
<tr>
<td>MoveTo VATarget Location</td>
<td>ac6</td>
<td></td>
</tr>
<tr>
<td>Pay VATarget ForProduct</td>
<td>ac7, ac8</td>
<td></td>
</tr>
<tr>
<td>MoveTo APA Location</td>
<td>ac9</td>
<td></td>
</tr>
<tr>
<td>ReportTo UAA</td>
<td>ac10</td>
<td></td>
</tr>
</tbody>
</table>

The messages that the MCA exchanges with the YPA, VA, and UAA agents during its lifecycle, reported in the MABD, are implemented through external events in the ADSC; the association between messages and events is reported in Table 2 for the interactions with YPA and VA.

<table>
<thead>
<tr>
<th>MABD MESSAGE</th>
<th>SENDER</th>
<th>RECEIVER</th>
<th>ADSC EVENT</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Request, VendorsList, RDF)</td>
<td>MCA</td>
<td>YPA</td>
<td>YPAListRequest</td>
</tr>
<tr>
<td>(Inform, VendorsList, RDF)</td>
<td>VA</td>
<td>YPA</td>
<td>YPAListInform</td>
</tr>
<tr>
<td>(Query, OfferPrice, RDF)</td>
<td>MCA</td>
<td>VA</td>
<td>OfferPriceQuery</td>
</tr>
<tr>
<td>(Inform, OfferPrice, RDF)</td>
<td>VA</td>
<td>MCA</td>
<td>OfferPriceInform</td>
</tr>
<tr>
<td>(Request, Payment, RDF)</td>
<td>MCA</td>
<td>VA</td>
<td>PayForRequest</td>
</tr>
<tr>
<td>(Inform, Payment, RDF)</td>
<td>VA</td>
<td>MCA</td>
<td>PayForInform</td>
</tr>
</tbody>
</table>

### 3.3. Simulation Phase

The simulation phase, supported by MASSIMO, allows for the validation of the system’s requirements, the behavior of each agent type and the related agent interactions, and the evaluation of two specific performance indexes:

- the **Buy Task Completion Time** \( T_{BTC} \), which is defined as \( T_{BTC} = T_{CREATION} - T_{REPORT} \) where, \( T_{CREATION} \) is the creation time of the MCA and \( T_{REPORT} \) is the reception time of the MCA report;  
- the **Probability of Successful Buy** \( P_{SB} \), which is defined as the probability of successfully buying a desired product within the e-Marketplace.

In particular, the simulation scenario was set up as follows:

- Each stationary agent (UAA, APA, YPA, VA, BA) executes in a different agent server.

- Agent servers are mapped onto different network nodes which are completely connected through links having the same characteristics and modeling the communication delay \( d \) as a lognormally distributed random variable.

Moreover, to compare the simulation results obtained for the evaluated performance indexes with the results of well defined analytical models, the simulated e-marketplace was a quite simple e-marketplace in which we supposed that each VA is reachable from any YPA and sells the same set of products, each product is always offered by a VA at a fixed price, which is an integer number uniformly distributed between a minimum \( (P_{MIN}) \) and a maximum \( (P_{MAX}) \), and the user is willing to pay, for a desired product, a maximum price \( P_{MAX} \) which is an integer value between \( P_{MIN} \) and \( P_{MAX} \).

Given the above described scenario, the evaluation of the \( T_{BTC} \) performance index is focused on an MCA adopting a searching policy (SP) of the ALL type and a buying policy (BP) of the MP type (see Table 3), moreover it is supposed that \( P_{MAX} = P_{MAX} \) so always guarantees a successful purchase at the best price.

The results, obtained adopting a YPA organization in which the YPAs are logically connected as a binary tree, are reported in Figure 13 with \( N_{YPA} = \{10, 100\} \) and varying \( N_{VA} \), where \( N_{YPA} \) is the number of the YPA agents and \( N_{VA} \) is the number of the VA agents. The simulation results agree with the results obtainable applying the analytical model reported in (Wang et al. 2002) and confirm that the PCA, due to its parallel dispatching mechanism, outperforms the ICA when \( N_{VA} \) increases.

<table>
<thead>
<tr>
<th>Searching Policy (SP)</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>ALL</td>
<td>All YPA agents are contacted</td>
</tr>
<tr>
<td>PA-FORMAL</td>
<td>A subset of YPA agents are contacted</td>
</tr>
<tr>
<td>OS-ONE-SHOT</td>
<td>Only one YPA agent is contacted</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Buying Policy (BP)</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>MP-Minimum Price</td>
<td>The MCA first interacts with all the VA agents; then, it buys the product from the VA offering the best acceptable price</td>
</tr>
<tr>
<td>FS-First Shot</td>
<td>The MCA interacts with the VA agents until it obtains an offer for the product at an acceptable price, then, it buys the product</td>
</tr>
<tr>
<td>FT-Fixed Trials</td>
<td>The MCA interacts with a given number of VA agents and buys the product from the VA which offers the best acceptable price</td>
</tr>
<tr>
<td>RT-Fixed Trials</td>
<td>The MCA interacts with a random number of VA agents and buys the product from the VA which offers the best acceptable price</td>
</tr>
</tbody>
</table>

On the basis of the assumptions made for the simulated e-marketplace, \( P_{SB} \) can be easily calculated as follows:

\[
P_{SB} = \left[ (P_{MAX} - P_{MIN}) / (P_{MAX} - P_{MIN} + 1) \right]^2, \quad \text{where: V is the number of VA agents contacted by the MCA for buying the product, } P_{MAX} - P_{MIN} \text{ represents the number of prices that exceed } P_{MAX} \text{ (i.e., that are not acceptable for the user), whereas } P_{MAX} - P_{MIN} + 1 \text{ represents the number of all the possible prices for the product. V depends on the BP adopted by the MCA; in particular: if BP is of the MP type or of the FS type } V = N_{VA}; \text{ if BP is of the FT type } V = F_{FT} \times N_{VA} / 2 + 1 \text{ as in the simulations the MCA always performs } N_{VA} / 2 + 1 \text{ trials; if BP is of the RT type } V \text{ belongs to the range } [1...N_{VA}] \text{. The values of } P_{SB} \text{ calculated both analytically and through simulation for each defined BP and with } P_{MAX} = 200, P_{MIN} = 100, P_{MAX} = 100, \text{ and } N_{VA} = 100, \text{ are reported in
Figure 13: Evaluation of $T_{BTC}$ for an MCA with $SP=ALL$, $BP=MP$, $N_{YPA}=[10, 100]$ and variable $N_{VA}$

4. CONCLUSION

This paper has proposed and exemplified through a case study an agent-orientated simulation-driven development process obtained by enhancing PASSI with a simulation step based on a simulation methodology centered on Distilled Statecharts and related tools. The resulting process represents a novel contribution to the AOSE research area as it is a new tool which promotes experimenting with the design and the simulation of complex MASs to support the development of higher-quality agent-based software systems. Currently our research efforts are geared at applying the simulation-driven development process for the construction and analysis of self-organizing MASs.

REFERENCES


AGENT & ARTIFACTS FOR SYSTEMS BIOLOGY:
TOWARD A FRAMEWORK BASED ON TuCSOn

Sara Montagna
Alessandro Ricci
Andrea Omicini
DEIS, ALMA MATER STUDIORUM—Università di Bologna
Via Venezia 52, Cesena, Italy
E-mail: sara.montagna@studio.unibo.it, {a.ricci, andrea.omicini}@unibo.it

ABSTRACT

Multiagent systems (MASs) are seemingly a promising way to model and simulate biological systems and networks, going toward a system-level understanding of such complex systems, as promoted by Systems Biology. In this paper we propose an agent-based framework based on the notion of agents and artifacts, implemented on top of the TuCSOn agent coordination model and infrastructure. The approach promotes the engineering of distributed simulations, where biological networks are modelled as multiagent systems. There, agents—representing active biological components such as proteins and enzymes—interact by means of tuple centres—representing the environment that enables and mediates the interaction of such components.

INTRODUCTION

Multiagent systems (MASs) have been widely recognised as a good approach for modelling and simulating complex systems (Epstein and Axtell 1996). As such, they can be seen as a promising choice also for modelling and simulating biological complex systems, in particular those scenarios in which traditional approaches—such as the one based on on Ordinary Differential Equation (ODE)—fail. Typically, in such scenarios interaction inside the system and between the system and its environment plays a fundamental role and must be explicitly taken into account in order to understand the overall system behaviour, typically emergent, along with related effects concerning non-linearity, stochastic phenomena, feedbacks, and so on.

Several research studies coming from different disciplines are pointing out the importance of such aspects for understanding living systems. A foremost and recent example is given by Systems Biology (Kitano 2002), which collects approaches that aim at understanding biology at the system level. Agent-based approaches provide a natural way to model such aspects. In literature, we can identify two basic families for such approaches: either (1) MASs with relatively complex, large-grained agents, typically interacting through direct communication and without a notion of agent environment, or (2) agent-based approach based on very simple homogeneous agents—similar to cellular automata—with an explicit notion of computational environment, with which they explicitly interact.

In this paper we introduce an agent-based approach to model and simulate biological networks, characterised by large-grained agents as in the first case, but with an explicit notion of environment, as in the second case. The approach is based on the notion of artifact, used to model agent environments, implemented on top of TuCSOn, a coordination model (and infrastructure) exploited in the context of multiagent system coordination.

Our general objectives are:

- From a scientific point of view, defining a computational model for Systems Biology based on agent and artifact concepts, with artifacts realised in particular on top of TuCSOn tuple centres. The idea is to exploit the peculiar features of such a model to set up virtual experiments in which scientists can (i) simulate the behaviour of biological systems by executing the corresponding MAS, and (ii) both observe and influence system evolution, with also the chance of interacting with the system while it is running.
- From the engineering point of view, defining a general-purpose distributed platform to be reused for different kinds of experiments, capable of running distributed simulations, exploiting multiple computing machines linked through a network.

The remainder of the paper is organised as follows. In the next section, we introduce some key points of Systems Biology, starting from the main aspects that characterise biological systems as complex systems and of biological networks in particular; Thereafter, we first introduce a modelling framework for biological networks based on agents and artifacts, then we describe a computational model based on TuCSOn, taken as a concrete model to implement artifacts. Finally, we conclude the paper by briefly discussing ongoing and future works.
Biological Systems as Complex Systems

Generally speaking, biological systems exhibit a number of key properties that characterise them as complex systems, namely:

hierarchy — structures and processes in biological systems are organised in hierarchical highly-structured levels, each one exhibiting a different kind of complexity (Dhar et al. 2004): sequences, molecules, pathways (such as metabolic or signalling), networks as collection of cross-interacting pathways, cells, organs;

interaction — each level is characterised by the interaction of a set of components inside some kind of environment. The interaction can occur both directly among the components, and between the components and their environment. Such interactions involve some kind of information exchange, where the information can be in the form of energy and material, frequently creating both negative and positive feedback loops;

emergent behaviours — each level exhibits forms of self-organisation among the components, resulting in the development of emergent behaviours, whose existence does not depend, cannot be predicted and cannot be explained from the knowledge about the individual components. In other words, the emergent properties cannot be readily explained in terms of the properties of individual components, just as systems are not the mere sum of their individual components (Dhar et al. 2004);

stochastic phenomena — stochastic effects play a crucial role at various levels of the hierarchy, in particular at those in which even the overall macro-behaviour can be influenced and determined by the interaction of a relatively small number of components. It is the case, for instance, of signal transduction pathways, involving a small number of particles (Chee Meng et al. 2004);

non-linearity and chaotic behaviour — the type of interaction among the components often results in systems exhibiting a non-linear behaviour, i.e. systems that are able to react in quite different manners to similar condition or small variations occurring in their environment. According to several researchers, this is an essential and distinguishing feature of living systems. Such interactions can even result in chaotic behaviours, which, according to several studies, seems to be essential in several contexts for keeping a good functioning of the overall system.

These aspects indeed make the analysis and comprehension of biological system dynamics an incredibly challenging task, for which traditional mathematical modelling approaches proved to be inadequate.

The Systems Biology Approach

The study of biological systems as complex systems, with a strong focus on the system level and on the interaction dimension, is among the key points of Systems Biology (SB), recently raised in bio-informatics contexts as innovative way of doing biology. SB promotes a more systemic and informatics-oriented way to conceive models, methodologies, and simulations of biological systems. Following Kitano (Kitano 2002), the aim of SB is understanding biology at the system level, moving from the understanding of the individual cellular components—which is the main aim of traditional approaches, described as reductionist—to the understanding of how these components interact as systems to produce the observed emergent behaviour, and how all the pieces are integrated and assembled together so as to get a systems view of biology.

This objective leads to focus more on the dynamics of cellular and organismal function rather than on the characteristic of the isolated parts of a cell or organism. For the purpose, we need to (a) obtain a massive amounts of data about whole biological systems via high through-put experiments; (b) build, with such data, a science of the principles of operation of biological systems, based on the interactions between components (Cardelli 2005).

The main investigating tool at the core of the SB approach—and of computational Systems Biology (CSB) in particular—are informatics systems (including network of computing and software systems), which provide the necessary means to create computational models of biological systems and of their environment (in-silico biology), and to observe and control the dynamic emergent behaviours of the systems through simulations.

Such computational models typically promote bottom-up methodologies and techniques for modelling and simulating systems, in contrast to top-down ones provided by traditional approaches. Top-down approaches are based on mathematical models created starting from the knowledge that scientists have about the overall system: how it behaves, what phenomena are linked to its behaviour, what are the general laws that rule such behaviour. Models are typically based on the identification of a basic set of variables representing key properties of the system behaviour—for instance: the concentrations of some type of molecules—and then by a set of ordinary differential equations (ODE) that describes how properties change along time and space dimensions. Such approaches typically require some strong continuity hypotheses to be raisable, which often are not satisfied in the biological systems under investigation. As a main example, they assume that the concentrations would change in a continuous and deterministic way, along quite long periods of time. This assumption is feasible only if we consider systems with a very high number of components, interacting in quite large environments, conditions that are not satisfied in most of
the intra-cellular scenarios that are investigated (Chee Meng et al. 2004).

Conversely, bottom-up approaches, which are frequently used to model and simulate complex systems in general, start from the knowledge about the individual components making the overall system, how they locally interact, and what is their structure and behaviour. A biological system is then described essentially in terms of the laws that locally characterise component interactions, with a level of abstraction that makes the model more adherent to what happens in nature.

**Biological Networks**

By adopting the SB point of view, different levels of biological systems can be identified constituting a hierarchy, each involving some different kind of interacting entities:

- interaction between molecules inside a *compartment*, i.e. a functionally distinct region of a cell enclosed by a membrane;
- interaction between compartments in a cell
- interaction between cells of a system (e.g. immune system)
- interaction between systems (immune, nervous, endocrine)

In this paper we focus on the first one, i.e. intra-cellular interaction and processes, which can be categorised in two main kinds of pathways or networks: *metabolic pathways* and *signalling pathways* (Dhar et al. 2004).

Metabolic pathways contain a series of biochemical reactions in which the product of one reaction serves as the substrate of the next reaction. These pathways control the production, transformation and consumption of energy in cells, and are usually self-sustained and self-sufficient.

Signalling pathways are the biochemical pathways regulating the flow of information within and between cells, including the ability to propagate information across different time scales and to function as switches and oscillators. Thanks to these pathways the cell is allowed to respond and to adapt to an ever-changing environment. In fact, context-dependency is actually the key feature of signalling pathways.

If traditional mathematical approaches (such as flux balance analysis) have proved to be quite effective in modelling and simulating metabolic pathways, the same does not hold for signaling pathways. The mechanisms underlying these complex behaviours involve many interacting components and cannot be understood by experiments alone. No adequate mathematical models are known for analysing such pathways.

**AGENTS & ARTIFACTS FOR SYSTEMS BIOLOGY**

As pointed out by Systems Biology, the development of new computational models of biochemical networks, integrating existing experimental data, will be fundamental in next years (and perhaps decades) to understand, model, control and re-create biological systems (Finkelstein et al. 2004). Several innovative computational approaches have been developed recently in research literature, both from researchers belonging to biology contexts—such as Shimizu and Bray (2001)—and from researchers in disciplines more related to computer science and engineering, and bioinformatics—examples are approaches based on process algebra (Regev et al. 2001, Curti et al. 2004, Regev et al. 2004), Petri Nets (Peleg et al. 2002), reactive systems (Fisher et al. 2005), and even agents and MASs (Khan et al. 2003, Querrec et al. 2003, Pogson et al. 2004, Gonzalez et al. 2003, Corradini et al. 2005, Borghoff et al. 1996). In the following we introduce an agent-based computational model recently developed in the context of complex software systems engineering, which we consider effective for modelling some crucial aspects of biological networks and systems in general.

**Agent and Artifact Abstractions for Modelling Complex Systems**

Agent-based and multiagent systems have already been recognised as a promising approaches for modelling and simulating biological systems. Generally speaking, existing approaches can be subdivided in two main categories:

- MAS-based models that adopt intelligent or cognitive agents, interacting through direct communication, typically based on speech act-like models. An example is given by Khan et al. (2003), where a MAS based on large-grained agents is used for the quantitative simulation of biological networks. These approaches draw their inspiration or directly adopt models and systems used for engineering real software systems.
- Agent-based approach based on very simple homogeneous agents interacting in grid-like environments (Epstein and Axtell 1996).

These approaches have been created specifically in the context of simulation-based systems, and share several aspects with other related models, such as cellular automata.

The approach proposed in this paper, on the one side is based on a notion of agent as depicted for the first category. Accordingly, agents can be defined as pro-active entities situated in some kind of environment, where they autonomously execute some kind of activity toward the achievement of some goal. MASs introduce a social dimensions, modelling systems as set of set of agents situated in a shared environment interacting according to various kinds of social models.
(coordination, cooperation, competition and so on). Such an approach is based on a full distribution and encapsulation of the control: each agent encapsulates a state, a behaviour and the control of the behaviour, that is, of its activities. Different computational models can be adopted for defining agent behaviour, according to the complexity of its activities: from simple reactive models to cognitive models, adopted by the so-called intelligent agents (Wooldridge and Jennings 1995).

Then, differently from existing agent-based approaches of the first category—and more similarly to approaches found in artificial like contexts—we introduce a computational model in which also the agent environment is represented as a first class entity, and such an entity is fundamental for enabling forms of interaction among agents different from direct communication, such as mediated interaction, which can play a key role in the overall system coordination. We exploit the notion of artifact as basic abstraction to model and structure agent environment (Omicini et al. 2006, Ricci et al. 2006). Artifacts can be generally defined as objects—as synonym of entities—encapsulating some kind of function, which agents either individually or collectively exploit in the course of their activities, analogously to artifacts as found in human society (where humans play the role of agents). Artifacts can be used to represent either the target of agent activities—with agents constructing, manipulating artifacts—or “tools” that agents use to help and sustain their activities. A simple example of artifact is given by a blackboard, as a kind of coordination artifact, designed to provide coordination functionality that agents can exploit in their collective activities. An agent interacts with an artifact through its interface, as a set of operations conceived by the artifact designer to let agents access and exploit artifact functionalities.

Artifacts then can be used as the basic abstraction to model such parts of a system that are not suitably captured by the agent abstraction: a primary example is given by the agent computational environment itself, in particular those parts that are specified and under control of MAS designers.

Modelling and Simulating Biological Networks

The general idea of this paper is to model and simulate biological networks and systems as multiagent systems exploiting the agent and artifact abstractions to represent different aspect such networks.

On the one side, we use agents to model any macromolecular component with a complex structure and behaviour, such as proteins, referred here as active bio-components. Agents are designed and programmed so as to simulate the behaviour of such components, modelled as autonomous stateful entities interacting with the bio-chemical environment where they are situated, acting in parallel, asynchronously. The behaviour of the active bio-components is then modelled in terms of agent activities or tasks, composed by set of actions with which the agents perceive and influence environment by consuming and producing molecules, and then changing their internal state. For instance, the computational behaviour of an agent representing a protein models how the protein changes its state according to the interaction with other molecules, and the interactive behaviour of protea sites—binding and unbinding molecules—as forms of actions and perceptions. So, the chemical reactions that characterise such bio-component can be used to design agent behaviour, modelling also important aspects such as reaction speed. Sensors and effectors of the agent represent the sites of the bio-components, through which it interacts with its environment, in our case with artifacts.

On the other side, artifacts are used to represent the biochemical environment (or compartment) that makes it possible for agents to survive and interact, both providing the operations to produce / consume molecules, and encapsulating and applying the rules that define and characterise such interaction. Such rules are not under the control of the individual interacting entities but are defined at a system level, and are essential to establish the overall system evolution. Among the responsibilities charged upon artifacts we count: (a) collecting and managing the set of molecules inside a compartment, as the resources that are shared and concurrently produced / consumed by the active bio-components as agents; (b) establishing dynamically which interactions can take place, i.e. which bio-components can consume / produce which molecules, according to what is known from the theoretical models, and taking into the account stochastic aspects. Actually, the artifact can be used also to embed and apply those chemical reactions that are not in charge to any specific agent.

A FRAMEWORK BASED ON TuCSoN

Given the conceptual background based on agents and artifacts, we exploit the TuCSoN coordination model and infrastructure to setup a general purpose framework for creating agent-based in-virtuo distributed experiments, in which biological systems / networks are modelled and simulated, starting from simple and well-known cases.

The TuCSoN Model

An example of concrete model and technology supporting the agents and artifact conceptual model is given by the TuCSoN agent coordination infrastructure (Omicini and Zambonelli 1999).

TuCSoN (Tuple Centres Spread over the Network) has been ideated in the software engineering context, as a coordination model and infrastructure to be exploited for engineering complex agent-based software systems. TuCSoN provides tuple centres as coordination artifacts that agent can use to
TuCSoN infrastructure makes it possible to exploit tuple centres in a distributed and network environment, by providing services to access and use tuple centres that are spread over the nodes of the network (Internet)—wherever the TuCSoN infrastructure has been installed. A TuCSoN node can contain any number of tuple centres, each one identified by a logic name. A software system engineered upon TuCSoN consists then by a MAS—adopting eventually different models and technologies for implementing the individual agents—distributed among some nodes and exploiting tuple centres, possibly located in different nodes, to communicate and coordinate.

In this paper we exploit TuCSoN on the one side as source of abstractions for defining the computational model for biological networks, and on the other side as an infrastructure and a technology for engineering distributed simulations, based on a general purpose framework for creating agent-based in-vitro distributed experiments.

A Simple Computational Model based on TuCSoN

By exploiting TuCSoN, we can keep the separation of concerns discussed in previous section, using artifacts—tuple centres in this case—to model the compartment and agents for representing active bio-components.

A compartment is modelled by a single tuple centre, in which: (a) we keep track of the the set of molecules inside the compartment as logic tuples contained by the tuple centre; (b) agents (active bio-components) can produce / consume (bind) molecules by executing suitable out and in operations on the tuple centre; (c) the rules establishing which interactions can take place are encapsulated in the tuple centre behaviour and encoded then in the ReSpeCT language.

In particular, as far as point (a) is concerned, we use tuples of the kind

\[
\text{molecule} \left( \text{Type}, \text{Number} \right)
\]

A tuple of such a type means that the compartment contains a number of molecules of type \text{Type}. \text{Type} is a simple name, used to identify the kind of molecules involved in the system: for example, atp, glucose. For instance a tuple \text{molecule} (\text{atp}, 1000) indicates that in the compartment there are 1000 molecules of ATP.

As far as point (b) is concerned, by using TuCSoN agent sensor and effectors—i.e. the sites of the bio-components—are realised through \text{in} and \text{out} operation, which agents invoke on the tuple centre. In particular, an agent binds molecules to its current set of free sites by executing an \text{in} operation of the kind

\[
\text{in} \left[ \text{molecules} \left( \text{MolTypes}, \text{Affinities}, \text{Result} \right) \right]
\]

where \text{MolTypes} is the list of the types of molecules that the agents can consume, one for each free site; \text{Affinities}...
is the list of the affinity coefficients associated to each molecules type—so one type for each site—, representing a measure of the probability of binding such molecule type:

**Result** is a list representing the number of molecules consumed for each type (site) specified in MolTypes, as result of the interaction. As an example, an agent executing an operation

```
in(molecules([atp,glucose],[0.5,0.8],Result))
```

has two free sites, which can be bound respectively to an ATP molecule and a glucose molecule, with affinity respectively of 0.5 for ATP and 0.8 for glucose. The operation succeeds if (when) one molecule is actually consumed from the compartment and bound to one of the site. As an example of tuple returned by the operation

```
molecules([atp,glucose],[0.5,0.8],[1,0])
```

indicating that one molecule of ATP has been bound to the first site.

Analogously, the production of molecules is realised by means of a simple **out** operation of the kind

```
out(molecules(MolTypes,Quantities))
```

where MolTypes has the same structure as before, and Quantities is a list of integers indicating the number of the molecules of the specified type must be produced. For instance, by executing a

```
out(molecules([glu,adp],[1,1]))
```

the agent produces in the compartment a molecule of glucose and a molecule of ADP.

Finally, as far as point (c) is concerned, the tuple centre is programmed so as to update the set of molecules according to agent operation requests on the one side, and to select the interactions that will occur on the other side. That is, given the overall set of agent requests (molecules that can be consumed), the tuple centre establishes which ones will be served, from time to time. For the purpose, different kinds of models can be experimented, implementing strategies of different complexity: for instance, the approach currently adopted takes into account the concentrations of the molecules and the affinity specified by agents to select—using stochastic laws—which interactions will occur—that is which molecules will be bound by agents. From the overall MAS point of view, such strategies represent the laws that rule agent cooperation and competition in consuming and producing molecules.

### An Example: Glycolysis

First experiments have been developed by considering glycolysis, a well-studied metabolic pathway occurring in citosol compartment.

The overall process counts ten steps in cascade, in which the product of a stage is the substrate needed for the next stage: the net result is the transformation of 1 molecule of glucose into 2 molecules of pyruvate, consuming two ATP during the process and producing, at the end, 4 ATP and 2 molecules of NADH. Each stage is carried on by some kind of enzymes, who mainly play the role of catalysts, consuming substrates and generating products. Actually, three of them—namely hexokinase (HK), phosphofructokinase-1 (PFK-1) and pyruvate kinase (PK)—play also another crucial role, as controllers of the overall process. Through allosteric regulating mechanisms, they act as valves of the glycolytic process, guaranteeing that globally on the one side there would not be an excess of products (ATP and pyruvate), on the other side that the speed of overall process would be adequate to produce the energy (ATP) needed for the cell to work.

By adopting the modelling approach presented in the paper, enzymes—controllers in particular—are modelled as agents and the citosol, the environment where the enzymes act, is represented by a tuple centre. Here we focus on the modelling of the PFK-1, represented by the EnzPFK agent: the others agent are analogous. The enzyme activity, as a catalyst, is to execute the following reaction:

\[
\text{F6P} + \text{ATP} \xrightarrow{\text{phosphofructokinase}^{-1}} \text{FBP} + \text{ADP}
\]

where F6P is fructose-6-phosphate and FBP is fructose-1,6-biphosphate.

As a valve, such activity changes according to its state. It has four different states—**ready**, **active**, **inhibited**, and **reactive**—, which changes according to the interaction that the enzyme has with its environment. In each state— which corresponds to a different physical structure and spatial configuration—the enzyme exposes different sites for binding molecules. In particular, the enzyme has five sites of three different kinds: two sites for binding substrate molecules—fructose-6-phosphate and ATP—, two sites for binding molecules that act as activators—both ADP—and one site for binding molecules that act as inhibitors, that are ATP.

The overall behaviour of the agent is briefly described as follows. In the **ready** state, the enzyme has all the sites free to bind molecules, resulting in an **in** of the type:

```
in:molecules([f6p, atp, atp, adp, adp], [0.5,0.5,0.1,1,1],Result))
```

If it binds both substrate molecules, it becomes **reactive**; if it binds ATP in the inhibitor state, it becomes **inhibited**; if it binds ADP, it becomes **active**.

In the **active** state the enzyme has a greater affinity with substrate and then a greater probability to bind such molecules. The free sites are the ones for binding substrate molecules and for inhibition. The resulting **in** is

```
in:molecules([f6p, atp, atp], [1,1,0.1],Result))
```

As in the previous case, if it binds both substrate molecules, it becomes **reactive**; if it binds ATP in the inhibitor state, it becomes **inhibited**.
In the reactive state, the enzyme transforms the substrate molecules in products, taking a certain amount of time (properly scaled) computed according to the speed of the reaction and stochastic effects, and then it releases the products in the citosol by means of an out of the kind

\[
\text{out}(\text{molecules}([\text{fhp, adp},]1,1))
\]

In the inhibited state, the agent interacts with the environment only by means of sites to be reactivated. The resulting in is:

\[
\text{in}(\text{molecules}([\text{adp, adp},]1,1),\text{Result})
\]

When both the sites have bound an ADP molecule, then it becomes ready again. It is easy to see that the more the concentration of ADP is high, the more the probability to bind ADP is high and the agent can start again to play its catalysing activity.

The tuple centre playing the role of citosol environment is programmed so as to behave as follows: (a) It updates the set of molecules (represented by the tuples molecule) accordingly to the action of the agents, consuming and producing molecules. In particular, it reacts to out(molecules(MolTypes,Quantities)) actions and updates all the tuples molecule(T,Q) for which T is in the MolTypes list, adding the value in Quantities to Q. (b) It keeps track of the agent requests (done with in(molecules(Mol,Aff,Res) actions) and generates the tuples that satisfy such requests according to the availability of the molecules requested, the affinity specified in the request and stochastic effects in general. Such a behaviour is encoded as a set of reactions in ReSpecT language.

**DISCUSSION**

The adoption of the agent and artifact abstractions to model biological systems—supported by an infrastructure such as TuCSoN—makes it possible to scale quite well with the complexity of the systems, from both the conceptual and engineering point of view.

From a conceptual point of view, by means of the agent abstraction we can model interacting entities specifying a behaviour that can range from simple stochastic behaviour (as in the case of PFK-1 enzyme in the example) up to entities with an “intelligent behaviour”, encapsulating more complex strategies for choosing actions and reacting to environment stimuli. This is particularly useful when depending on the specific biological system—it is more natural for researchers and scientists to model the behaviour of the components starting from the knowledge they have about what is the role (objective or goal) of such components inside the systems, instead of specifying how they achieve such objectives (in terms of detailed interactions).

From an engineering point of view, TuCSoN naturally supports distributed simulations, in which agents (interacting through the same tuple centre) can be distributed over multiple machines, as well as tuple centres, representing distinct compartments.

By virtue of such a flexibility, it will be interesting to explore the approach for modelling and simulating other biological networks at a different level, such as artificial immune systems, comparing the results with existing approaches (Stepney et al. 2005, Celada and Seiden 1992).

**ONGOING & FUTURE WORK**

Starting from the computational model presented in this paper, ongoing works are focussing on setting up a suitable application environment for flexibly executing distributed simulations, testing the validity of the model. Such application environment is based on the TuCSoN technology, available on Java platform as an open source project.¹

As previously mentioned, the objective is to create a virtual laboratory distributed on multiple machines, on top of which executing distributed simulations engineered as MASs, with agents interacting through one or multiple tuple centres, according to the specific experiments. The idea is to create an environment not only for executing the simulations and observing dynamically the results from the simulation run, but also to execute online experiments, in which scientists—supported by suitable tools—could interact with the ongoing systems, for instance changing the structure of the systems, both by introducing / removing agents (active components, such as proteins, enzymes) or by changing the content of the tuple centres (that is, changing the molecules quantities of the compartments).

As test cases, we will consider some well-studied biological pathways, such as the glycolysis metabolic pathway as briefly discussed in the paper, and the MAPK (Mitogen-Activated Protein Kinase) transduction pathway, which has a key role in cell growth and cell cycle.

**REFERENCES**


¹[http://tucson.alice.unibo.it](http://tucson.alice.unibo.it)


Acknowledgment

We would like to thank the reviewers for their precious comments, which have been very helpful to identify further issues to investigate, and our colleagues bioengineers and biochemists at DEIS for their support and cooperation.
SIMULATING MINORITY GAME WITH TuCSoN

Enrico Oliva
Mirko Viroli
Andrea Omicini
DEIS, ALMA MATER STUDIORUM—Università di Bologna
via Venezia 52, 47023 Cesena, Italy
E-mail: {enrico.oliva,mirko.viroli,andrea.omicini}@unibo.it

ABSTRACT

Minority Game is receiving an increasing interest because it models emergent properties of complex systems including rational entities, such as for instance the evolution of financial markets. As such, Minority Game provides for a simple yet stimulating scenario for system simulation.

In this paper, we aim at showing new perspectives in agent-based simulation by adopting a novel MAS meta-model based on agents and artifacts, and by applying it to Minority Game simulation. To this end, we adopt the TuCSoN infrastructure for agent coordination, and its logic-based tuple centre abstractions as artifact representatives. By implementing Minority Game over TuCSoN, we show some of the benefits of the artifact model in terms of flexibility and controllability of the simulation.

INTRODUCTION

Minority Game (MG) is a mathematical model that takes inspiration from the “El Farol Bar” problem introduced by Brian Arthur (Arthur 1994). It is based on a simple scenario where at each step a set of agents perform a boolean vote which conceptually splits them in two classes: the agents in the smaller class win. In this game, a rational agent keeps track of previous votes and victories, and has the goal of winning throughout the steps of the game—for which a rational strategy has to be figured out. Several researches showed that, although very simple, this model takes into account crucial aspects of some interesting complex systems coupling rationality with emergence: e.g. bounded rationality, heterogeneity, competition for limited resources, and so on. For instance, MG is a good model to study market fluctuation, as an emergent property resulting from interactions propagating from micro scale (agent interaction) to macro scale (collective behaviour).

As showed by Renz and Sudeikat (2005), a multiagent system (MAS) can be used to realise a MG simulation—there, BDI agents provide for rationality and planning. An agent-based simulation is particularly useful when the simulated systems include autonomous entities that are diverse, thus making it difficult to exploit the traditional framework of mathematical equations.

In this paper we proceed along this direction, and adopt a novel MAS meta-model based on the notion of artifact (Ricci et al. 2006). The notion of artifact is inspired by Activity Theory (Ricci et al. 2003): it represents those abstractions living in the MAS environment that provide a function, which agents can exploit to achieve individual and social goals. The engineering principles promoted by this meta-model makes it possible to flexibly balance the computational burden of the whole system between autonomy of the agents and the designed behaviour of artifacts.

In order to implement MG simulations we adopt the TuCSoN infrastructure for agent coordination (Omicini and Zambonelli 1999), which introduces tuple centres as artifact representatives. A tuple centre is a programmable coordination medium living in the MAS environment, used by agents interacting by exchanging tuples (logic tuples in the case of TuCSoN logic tuple centres). As we are not concerned much with the mere issues of agent intelligence, we rely here on a weak form of rationality, through logic-based agents adopting pre-compiled plans called operating instructions (Viroli and Ricci 2004).

By implementing MG over TuCSoN, we can experiment with flexibility and controllability of the artifact model, and see if and how they apply to the simulation—in particular, artifacts allow for a greater level of controllability with respect to agents. To this end, in this paper we show how the model allows some coordination parameters to be changed during the run of a simulation with no need to stop the agents: this can be useful e.g. to change the point of equilibrium, controlling the collective behaviour resulting by interactions propagated from the entities at the micro level.

The remainder of this paper is organised as follows. First, we introduce the general simulation framework based on agents and artifacts. Then, we provide the reader with some relevant details of the Minority Game. Some quantitative results of MG simulation focussing on system dynamics and run-time changes are presented, just before final remarks.
THE TuCSoN FRAMEWORK FOR SIMULATION

The architecture proposed for MAS simulation is based on TuCSoN (Omicini and Zambonelli 1999), which is an infrastructure for the coordination of MASs. TuCSoN provides agents with an environment made of logic tuple centres, which are logic-based programmable tuple spaces. The language used to program the coordination behaviour of tuple centres is ReSpeCT, which specifies how a tuple centre has to react to an observable event (e.g. when a new tuple is inserted) and has to accordingly change the tuple-set state (Omicini and Denti 2001). Tuple centres are a possible incarnation of the coordination artifact notion (Omicini et al. 2004), representing a device that persists independently of agent life-cycle and provides services to let agents participate to social activities.

In our simulation framework we adopt logic-based agents, namely, agents built using a logic programming style, keeping a knowledge base (KB) of facts and acting according to some rule—rules and facts thus forming a logic theory. The implementation is based on tuProlog technology\(^1\) for Java-Prolog integration, and relies on its inference capabilities for agent rationality. Agents roughly follow the BDI architecture, as the KB models agent beliefs while rules model agent intentions.

To coordinate agents we take inspiration from natural systems like ant-colonies, where coordination is achieved through the mediation of the environment: our objective is to have a possibly large and dynamic set of agents which coordinate each other through the environment while bringing about their goals.

Externally, we can observe overall system parameters by inspecting the environment, namely, the tuple centres agents interact with. In this way we can try different system behaviours changing only the coordination behaviour of the environment. Furthermore we can change, during the simulation, some coordination parameters (expressed as tuples in a tuple centre), programming and then observing the transition of the whole system either to a new point of equilibrium or to a divergence.

Three kinds of agents are used in our simulation: player agents, monitor agents and tuning agents (as depicted in Figure 1): all the agents share the same coordination artifact. The agent types differ because of their role and behaviour: player agents play MG, the monitor agent is an observer of interactions which visualises the progress of the system, the tuning agent can change some rules or parameters of coordination, and drives the simulation to new states. Note that the main advantage of allowing a dynamic tuning of parameters instead of running different simulations lays in the possibility of tackling emergent aspects which would not necessarily appear in new runs.

The main control loop of a player agent is a sequence of actions: observing the world, updating its KB, scheduling next intention, elaborating and executing a plan. To connect agent mental states with interactions we use the concept of action preconditions and perception effects as usual.

MINORITY GAME

MG was introduced and first studied by Challet and Zhang (1997), as a means to evaluate a simple model where agents compete through adaptation for finite resources. MG is a mathematical representation from ’El Farol Bar’ problem introduced by Arthur (1994), providing an example of inductive reasoning in scenarios of bounded rationality. The game consists in an odd number N of agents: at each discrete time step t of the game an agent i takes an action \(a_i(t)\), either 1 or \(-1\). Agents taking the minority action win, whereas the majority losses. After a round, the total action result is calculated as:

\[
A(t) = \sum_{i}^{N} a_i(t)
\]

\(^1\)http://tuprolog.alice.unibo.it
In order to take decisions agents adopt strategies. A strategy is a choosing device that takes as input the last \( m \) winning results, and provides the action (1 or -1) to perform in the next time step. The parameter \( m \) is the size of the memory of the past results (in bits), and \( 2^m \) is therefore the potential past history that defines the number of possible entries for a strategy.

The typical strategy implementation is as follows. Each agent carries a sequence of \( 2^m \) actions, called a strategy, e.g. \([+1, +1, -1, -1, +1, -1, +1, +1]\). The information on past wins is mapped on a natural number between 0 and \( 2^m - 1 \), which is used as position in the above sequence of the next action to take: for instance, if \([-1, +1, -1]\) is the past winning group, we read it as 010 (that is, 2), and accordingly pick the decision in position 2 inside \([+1, +1, -1, -1, +1, -1, +1, +1]\), that is -1.

Each agent actually carries a number \( s \geq 2 \) of strategies. During the game the agent evaluates all its strategies according to their success, and hence at each step it decides based on the most successful strategy so far. Figure 3 shows a typical evolution of the game.

One of the most important applications of MG is in the market models: Challet et al. (2000) use MG as a coarse-grained model for financial markets to study their fluctuation phenomena and statistical properties. Even though the model is coarse-grained and provides an over-simplified micro-scale description, it anyway captures the most relevant features of system interaction, and generates collective properties that are quite similar to those of the real system.

Another point of view, presented e.g. by Parunak et al. (2002), considers the MG as a point in space of a Resource Allocation Game (RAG). In this work a generalisation of MG is presented that relaxes the constraints on the number of resources, studying how the system behaves within a given range.

In a more recent paper, Renz and Sudeikat (2005) observe that MG players could be naturally modelled as agents with a full BDI model, and use a new adaptive stochastic MG with dynamically evolving strategies in the simulation.

**MG Performance**

In order to track the performance of an MG system, the most interesting quantity is variance, defined as \( \sigma^2 = [A(t) - \bar{A}(t)]^2 \); it shows the variability of the bets around the average value \( \bar{A}(t) \). In particular, the normalised version of variance \( \rho = \sigma^2 / N \) is considered.

Generally speaking, variance is the inverse of global efficiency: as variance decreases agent coordination improves, making more agents winning. Variance is interestingly affected by the parameters of the model, such as number of agents \( N \), memory \( m \) and number of strategies \( s \); in particular, the fluctuation of variance is shown to depend only on the ratio \( \alpha = 2^m / N \) between agent memory and the number of agents.

For large values of \( \alpha \)—the number of agents is small with respect to the number of possible histories—the outcomes are seemingly random: the reason for this is that the information that agents observe about the past history is too complex for their limited processing analysis.

When new agents are added, fluctuation decreases and agents perform better by choosing randomly, in this case \( \rho = 1 \) and \( \alpha \approx 1/2 \), as visible in the results of our simulation in Figure 4—the game enters into a regime where the loosing group is close to \( N/2 \), hence we might say coordination is performing well.

If the number of agents increase further, fluctuations rapidly increase beyond the level of random agents and the game enters into the crowded regime. With a low value of \( \alpha \) the value of \( \sigma^2 / N \) is very large: it scales like \( \sigma^2 / N \approx \alpha^{-1} \).

The results of other observations suggest that the behaviour of MG can be classified in two phases: an information-rich asymmetric phase, and an unpredictable or symmetric phase. A phase transition is located where \( \sigma^2 / N \) attains its minimum \( (\alpha_c = 1/2) \), and it separates the symmetric phase with \( \alpha < \alpha_c \) from an asymmetric phase with \( \alpha > \alpha_c \).

All these cases have been observed with the TuCSoN simulation framework described in next section.
THE SIMULATION FRAMEWORK

The construction of MG simulations with MAS is based on the TuCSoN framework and on TuProlog as an inferential engine to program logic agents. The main innovative aspect of this MG simulation is the possibility of studying the evolution of the system with particular and different kinds of agent behaviour at the micro level, imposed as coordination parameters which are changed on-the-fly.

Operating Instructions

Each agent has an internal plan, structured as an algebraic composition of allowed actions (with their preconditions) and perceptions (with their effects), that enables the agent to use the coordination artifact to play the MG. This plan can be seen as Operating Instructions (Violi and Ricci 2004), a formal description based on Labelled Transition System (LTS) that the agent reads to understand what its step-by-step behaviour should be. Through an inference process, the agent accordingly chooses the next action to execute, thus performing the cycle described in Section 2.

Operating instructions are expressed by the following theory:

```
firststate(agent(first,[[]]).
definitions(
def(first,[_],...),
def [main,[S]],
[act (out (play(X)), prep (choice(S,X))),
   per (in (result(Y)), eff (res(Y))),
   agent (main,[S])] )
);
```

The first part of operating instructions is expressed by term `first`, where the agent reads the game parameters that are stored in the KB, and randomly creates its own set of strategies.

In the successive part `main`, the agent executes its main cycle. It first puts tuple `play(X)` in the tuple space, where \( X = \pm 1 \) is agent vote. The precondition of this action `choice(S,X)` is used to bind in the KB X with the value currently chosen by the agent according to strategy S. Then, the agent gets the whole result of the game in tuple `result(Y)` and applies it to its KB. After this perception, the cycle is iterated again.

Tuple Centre Behaviour

The interaction protocol between agents and the coordination artifact is then simply structured as follows. First each agent puts the tuple for its vote. When the tuples for all agents have been received, the tuple centre checks them, computes the result of the game—either 1 or -1 is winning—and prepares a result tuple to be read by agents.

The ReSpecT program for this behaviour is loaded in the tuple centre by a configuration agent at bootstrap, through operation `set_spec()`. The following `ReSpecT` reaction is fired when an agent inserts tuple `play(X)`, and triggers the whole behaviour:

```
reaction(out (play(X)),
  in_r (count(Y)),
  Z is Y+1,
  in_r (sum(M)),
  V is M+X,
  out_r (sum(V)),
  out_r (count(Z))
).
```

This reaction considers the bet \( (X) \) counts the bets \( (Z) \) and computes the partial result of the game \( (V) \). When all the agents have played, the artifact produces the tuple `winner(R,NS,T1,T2,last/more)`, which is the main tuple of MG coordination.

```
reaction(out_r (count(X)),
  rd_r (numag [Num]),
  X:=r,Num,
  in_r (totcount(T)),
  P is T+1,
  rd_r (game(G)),
  in_r (sum(A)),
  out_r (sum(G)),
  rd_r (countsession(CS)),
  in_r (count(Y)),
  out_r (count(O)),
  % calculate variance
  in_r (gsum(SQ)),
  NSQ is A+A+SQ,
  out_r (gsum(NSQ)),
  % calculate mean
  in_r (totsum(R)),
  NewS is R*A,
  out_r (totsum(NewS)),
  rd_r (tuning1(T1)),
  rd_r (tuning2(T2)),
  out_r (winner(A,P,CS,T1,T2,G)),
  out_r (totcount(P))
).
```

![Figure 5: Interface of the Monitor Agent](image-url)
Figure 6: Variance of the System with Initial Parameters $N = 5$ and $m = 3$

The winner tuple contain the result of game $R$, the number of step (NS), two tuning parameters ($T1$ and $T2$) and one constant to communicate agents whether they have to stop or to play further (last/more). Figure 5 reports the graphical interface of the monitor agent that during its life-time reads the tuple winner and draws variance.

The simulation architecture built in this way allows for on-the-fly change of some game configuration parameters—such as the dimension of agent memory—with no need to stop the simulation and re-program the agents.

By changing the parameters, the tuning agent can drive the system from an equilibrium state to another, by controlling agent strategies, the dimension of memory, or the number of losses that an agent can accept before discarding a strategy. This agent observes system variance, and decides whether and how to change tuning parameters: reference variance is calculated by first making agents playing the game randomly—see Figure 4. The new value of parameters is stored in tuple centre through tuples tuning1($T1$) and tuning2($T2$), the rules of coordination react and update the information that will be read by the agents.

Simulation Results

The result of the tuned simulation in Figures 6 and 7 shows how the system changes its equilibrium state and achieves a better value of variance. In this simulation the tuning agent is played by a human that observes the evolution of the system and acts through the tuning interface to change the coordination parameters, such as threshold of losses and memory, hopefully finding new and better configurations. The introduction of the threshold of losses in the agent behaviour is useful when the game is played by few agents: these parameters enable system evolution and a better agent cooperative behaviour.

In Figure 6, the first phase of equilibrium is followed by a second one obtained by changing the threshold parameter $S = 5$. Finally, a third phase is obtained changing the dimension of the memory to $m = 5$.

Figure 7: System Evolution of the Variance in Figure 6

CONCLUSION

In this paper, we aim at introducing new perspectives on agent-based simulation by adopting a novel MAS meta-model based on agents and artifacts, and by applying it to Minority Game simulation. We implement and study MG over the TuCSoN coordination infrastructure, and show some benefits of the artifact model in terms of flexibility and controllability of the simulation. In particular, in this work we focus on the possibility to build a feedback loop on the rules of coordination driving a system to a new and better equilibrium state. Many related agent simulation tools actually exist: as this paper is a starting point, we plan to perform a systematic comparison of their expressiveness and features. In the future, we are interested in constructing an intelligent and adaptive tuning agent with a BDI architecture, substituting the human agent in driving the evolution over time of the system behaviour.

REFERENCES


MULTI-AGENT SYSTEMS
BIOMETRICS AUTHENTICATION AND REPUTATION BASED POLICIES 
IN A MAS SECURITY FRAMEWORK

V. Conti, G. Milici, F. Sorbello
Dip. Ingegneria Informatica
Università di Palermo
Viale delle Scienze Ed. 6
90128 Palermo, Italy
{conti, milici, sorbello}@unipa.it

S. Vitabile
Dip. di Biotecnologie
Mediche e Medicina Legale
Università di Palermo
Via del Vespro 90127 Palermo, Italy
vitabile@unipa.it

G. Pilato
I.C.A.R.
Italian National Research Council
Viale delle Scienze 90128 Palermo, Italy
g.pilato@icar.cnr.it

KEYWORDS
Agent ownership, biometric authentication, user reputation, multi-agent systems security.

ABSTRACT
Multi-agent systems expose users to risks related to lack of knowledge above interacting users. Agent ownership implies that a specific person or organization (the owner) is responsible for the agent's actions. Security requirements in the agent ownership have to protect their own resources from illegal accesses by unauthorized users. This paper describes a security framework to develop multi-agent systems preventing some challenging problems like: (i) safeguard of users (agent owners) against malicious users they could interact with; (ii) access restriction to protected resources. The first issue has been tackled using authentication process based on biometric features and creating a reputation-based trust system. The latter issue has been tackled by means of an access control mechanism based on policy files. To prove the effectiveness of the developed security infrastructure, added in the JADE-S platform, a distributed document retrieval system has been implemented.

INTRODUCTION

An e-Business agent community is a self-organized virtual space consisting of a large number of agents with a dynamic environment. Mobile agents can be owned by individuals or organizations. An agent owner can use his/her agent to carry out tasks to fulfill its own purposes. The agent services can be offered to individuals or organizations that don’t own specific agents. Agent ownership implies that a specific person or organization (the owner) is responsible for the agent's actions. Possible problems regarding security for users are: “requiring a service”, a service offered by a server could be not consistent with the published features (quality, functionalities, efficiency, cost), and “offering a service”, protection against malicious accesses and requests implies an efficient Access Control Mechanism. In a framework supporting agent ownership, agents are legal entities employed to bring their own principal into contractual relations with third parties. The authentication process establishes the identity of each owner and, consequently, of each agent. A policy, based on the previous identity, can determine the access level of an agent in e-Business systems, the permission to access to certain resources or perform certain tasks. The Agentcities Security Working Group has defined a set of security requirements identified for multiple agent platforms active in open distributed multi-domain networks. Among the application driven requirements, user authentication based on both invasive and non-invasive biometric features is suggested [1]. Security requirements in the agent ownership setting process are the identification and the reputation of the owner and the protection of the identity information carried out by an agent. As a consequence, in this paper a strong owners authentication process, a Reputation-based trust system to establish user honesty, the information encryption techniques and certificates based on X-security approach [5], have been treated. An unauthorized person, by stealing a trusted username/password pair, can gain access into a system and run malicious agents to perform unauthorized transactions. The original owner will be the legal responsible of these actions. In a mobile-agent framework, biometric based authentication systems can be used to improve the owner authentication process security [9]. As a consequence, agent and owner reputations are strictly related and they can be considered as a single entity. A network authority can temporarily or permanently suspend or revoke digital certificates to untrusted agents (users). In this paper, the mobile agent ownership issue is addressed by introducing biometric based systems in the user (owner) authentication process. A mobile agent declares its identity by showing its Identity Certificate, signed by a Certification Authority (CA). Using digitically-signed certificates with biometric information, the platform can be sure of the agent owner identity. In multi-agent systems, each agent is exposed to risks related to lack of knowledge about the honesty of other interacting agents. An agent requiring services must evaluate trustworthiness (honesty and goodness) of agent providing services. AGent Service Providers (AGSP) could provide a network infrastructure to allow people use agent-services offered in secure manner. Trust in computer systems security is based on access control. Trust management is tightly connected with creating, acquiring and distributing certificates, that are typically used in conjunction with security policies to determine whether to accept agent requests. In this paper a methodology based on reputation to establish which agents to trust, is used.
The proposed services and resources access control model use XML-based policy files. It joins the advantages of the three well-known access control methods [11][12]:

1. Role Based Access Control (RBAC). It consists of a set of grouping users with assigned roles. A role is a set of responsibilities, tasks and errands inside a group or organization. The roles are mapped in authorizations granted to users groups sharing a common role.

2. Mandatory Access Control (MAC). It consists of a set of grouping of users and resources into security levels, only users with the required security levels are allowed to access a certain resource.

3. Credential Based Access Control (CBAC): distributed access control management by signed certificates containing credentials.

The proposed access control mechanism supports the rights delegation. A subject performing a set of action, can, under certain conditions, grant the corresponding authorizations to other subjects trusted by himself.

In order to test the effectiveness of the proposed approach, the developed techniques have been implemented as an extension of the JADE-S platform [3]. In addition, the whole framework has been tested implementing the multi-agent document retrieval system proposed by the authors in [10].

BIOMETRIC FOR AUTHENTICATION

Mobile agent ownership requires reliable personal recognition schemes to determine and confirm the identity of an agent. Common authentication systems for multi-agent platform are based on username and password. A biometric based authentication process is based on two sequential phases [4]:

- enrolment: the system acquires an individual biometric information building a template;
- matching: the acquired biometric information is compared with the stored individual biometric information to determine whether or not they belong to the same person.

Different biometric features can be used in the user authentication process. Some of these features can be considered invasive biometrics (e.g. fingerprints, retina) and they may not be desirable as they infringe on privacy, others that are not unique or robust (e.g. hand geometry, voice) are less invasive.

User authentication systems performs the comparison one to one between the sensor acquired biometric features and the single related stored item.

REPUTATION AND TRUST

The problem of trust has many implications in an open distributed environment. In e-business transactions an agent needs to decide if another agent encountered in the network can be trusted or not.

In agreement with Yip and Cunningham [2], secure user authentication helps to sustain a trust model including agent ownership. Here, we distinguish two different types of owner:

- The authors are people or organizations that write programs to execute an agent;
- The senders are people or other entities that send agents to act on their behalf.

In the mobile agent identification issue, X.509v3 digitally-signed certificates, containing owner biometric information, can be adopted. Certificates are useful for agent authentication as well as to carry information about Agent and service reputation. A trusted multi-agent platform with agent execution tracing capabilities can decrease agent reputation for authors and senders whose agents are malicious or untrusted.

In multi-agent systems if an agent C trusts agent S with respect to service SV [7]:

1. C thinks that S is able to perform SV (ability)
2. C thinks that S will perform SV (availability)

Trust of C towards S with respect to SV is a precondition of the request of SV to S (delegation of SV to S). If C is not able to perform SV, it could delegate SV to S if the amount of trust towards S is “high enough” with respect to the “utility” (benefit) of SV [7]. That is, an agent decides to make a delegation only if the correlated risk is justified by the benefit it expects.

Reputation is a way to establish the amount of trust to grant to the others. Reputation is an aggregate value representing the amount of satisfaction of all the agents that have interacted, in the past, with the reputation target, that is a service/resource provided by a certain subject through one or more agents. The described reputation system associates a reputation value to each service provided by each user, allowing agents to realize about the trustworthiness of an agent providing services without having transaction with him.

THE PROPOSED ACCESS CONTROL FRAMEWORK

The framework analysis has been based on JADE-S. JADE-S is formed by the combination of the standard version of JADE with the JADE security plug-in [3]. JADE-S includes security features such as user/agent authentication, authorization and secure communication between agents into the same platform. However, neither is biometric authentication module provided nor digital certificates with extended information managed.

A fingerprint based authentication system for the JADE-S platform has been proposed by the authors in [9]. The whole platform as well as each single agent can be activated only by authenticated users. The activated agent will own a digitally-signed X.509v3 certificate containing owner personal information, owner authentication information (matching function and matching score) and the reputation ratings.

Our solution is based on a set of PKI techniques including certification authorities, private/public keys, and digitally-signed certificates. With more details, the multi-agent platform has been extended implementing a new Login Module (LM), an enhanced Security Certification Authority (SCA) [9] and a Reputation Manager Agent (RM) [13]. The LM is able to deal with username, password and fingerprint, the SCA is able to deal with digitally-signed X.509v3 certificates and the RM treat the reputation.
To trial the proposed approach a JADE-S multi-agent platform [3] based distributed document retrieval system was implemented [10].

The Developed Extensions

The Login Module
The Java Authentication and Authorization Service (JAAS) [6] allows:

- user authentication, to reliably and securely determine the user which is currently executing Java code;
- user authorization, to ensure they have the access control rights (permissions) required for the requested actions.

The developed Login Module manages the platform access system dealing with:

1. owner username and password;
2. a stored key containing owner credential;
3. owner biometric information.

The Login Module is depicted in Fig. 1; the module is composed by the following blocks:

- JAAS LoginContext: it is responsible for the authentication process. It reads a configuration file, named “jaas.config” and starts the appropriate LoginModule.
- Biometric Authentication: it verifies the user’s credentials and returns “true” or “false”. It checks also the username with the associated password, verifies the certificate authenticity, and performs the fingerprint matching.
- JAAS LoginModule: it creates a callback handler to acquire the user’s credentials. The credentials are sent to the E-CSAI Authentication Service that verifies them. If the verification is positive, the object session is generated.
- Session object: it contains a Principal Entity which is an authenticated entity having shared state for trusted multi-platform migration; a Public Credential (a public key); and a Private Credential (a private key).

Authentication system procedure is the following:

1. username and password request;
2. user data verification;
3. if the user is not registered (enrolment phase)
   3.1 a Registration Module starts requiring user personal data;
   3.2 user personal data acquisition;
   3.3 user biometric features acquisition;
   3.4 X.509v3 certificate and signed biometric information releasing;
4. if the user is registered (matching phase):
   4.1 signed biometrics data and certificate requests;
   4.2 owner biometric features acquisition;
   4.3 matching.

The triplet (username, password, biometric features) is adopted in order to have three different authentication items and to prevent Denial of Service (DoS) attacks for the authentication process.

![Figure 1. The Login Module and its components.](image)

The Security Certification Authority
Agent communications are managed with an extended version of the X-Security 2.0 package [5]. X-Security 2.0 supplies a secure model for inter-platform communication, opening several secure communication channels at the same time also with untrusted networks (differently from SSL).

X-Security 2.0 package has the following basic functionalities:

- a Security Certification Authority (SCA) – SCA is an independent agent which can temporally or permanently suspend, renew or revoke agents digital certificates. It is not a part of the platform (like AMS or DF);
- a Security Module (SM) – SM is an optional module that each agent adopts for secure communications. If no secure communications are required, the agent use the standard Agent Communication Channel.

The SCA has been extended adding the Login Module and a Registration Module (for the enrolment phase) above described, a Certificates and Keys Manager for X.509v3 certificates and finally a Reputation Manager Module (RMM) described in next session.

The Extensions field of the X.509v3 certificate contains:

- the name of the function to use in the matching phase. The function must be shared by the SCA;
- the minimum score needed for a positive match;
- the owner reputation ratings.

The developed SCA is able to deal with the above items implementing the related policies to grant or deny permission for performing certain actions. A mobile agent proves its identity to the platform by showing its Identity Certificate, signed by a SCA.

In Fig. 2, the agent platform structure with the emphasized SCA and Security Module is depicted.
Access Control Authority and Security Domains

Security Domain (SD) is a set of agents granting resources and services, based on access certificates conforming to CBAC, issued by a particular agent called Access Control Authority (ACA). For each SD there is one only ACA that manages access control for protected resources of SD agents [13]. Each certificate contains a list of roles, instead of a list of authorizations, like in traditional CBAC.

User authorization through roles, according to RBAC, allows users grouping and it is less error-prone than directly assigning authorizations separately [11]. Each role is mapped to a set of authorizations, expressed as Security Level. It allows resource grouping based on their degree of security. The greater is protection need, the greater is its security level.

The subject-role and role-authorization relationships are implemented by policy files. These files are read respectively by the ACA and by each agent providing services belonging to the SD managed by the ACA. Administrators have only to specify subjects-roles and roles-authorizations mappings, that are relatively constants [11].

CRAC (certificates) enables a very simple distributed access control management. In fact once an agent has got a (access or delegation) certificate, it can exhibit authorizations to other agents providing service without considering the physical location of ACA. In figure 3 the UML sequence diagram shows a typical sequence of interactions related to access to a protected service/resource.

Policy files

The proposed access control mechanism uses two kinds of policy files:

1. Files implementing subjects-roles mappings. They are examined by each ACA when it releasing a certificate to the requesting subject.
2. Files implementing roles-authorizations mappings. They are read by each agent to establish the authorizations of a subject exhibiting certificate.

We adopt XML based policy files, because they are easily readable by both human beings and software agents.

Rights delegation

If a role referenced in an access certificate is declared delegable, the owner is able to transfer (delega), by means of a delegation certificate, the authorizations corresponding to that role to another subject [13]. Delegation certificates are signed by their issuer and include the access certificate of the issuer. Consequently each agent providing services can establish if the issuer is allowed by the ACA to delegate the roles referenced in the certificate. Moreover, each role referenced in a delegation certificate can be declared delegable. In this case, the subject having the delegation certificate is allowed to re-delegate that role to another one through another delegation certificate. This certificate must include the original (the issuer’s) delegation certificate, so that agent providing services, examining the certificate chain, can establish the legality of the rights delegation.

The Reputation Manager Agent

In the implemented framework reputation is managed by a set o’ agents, called Reputation Managers (RM), which are started by the SCA reputation management module (RMM). RM cooperates to calculate, update and publish each reputation value. Reputation data is protected, when traversing through the agent network, by means of cryptographic techniques. At startup, the Reputation Manager Module (RMM) of each trust domain, executes several Reputation Managers agents (RM). Each time an user starts an agent requiring identity certificate, the RMM assigns the user to a RM [13]. Agents of each user interact only with their RM to submit feedbacks or to ask queries about the reputation of other agent. In such cases, the RM acts with respect of its agents, as a proxy towards the reputation system (the set of all the Reputation Manager). In fact, it forwards reputation queries to the RM responsible of request agent. Reputation Managers must be located on safe hosts.

Reputation updating is triggered by service satisfaction feedbacks received from other agents. Messages are exchanged between a subset of RMs determined by the internal reputation updating algorithm adopted. In particular,
a distributed version of the EigenTrust algorithm [8], is running on RMs:

1. The algorithm is executed by a limited and variable number of collaborating RMs.
2. For every user (subject) under the responsibility of a certain RM, the reputation related to each service is managed separately.
3. To provide a feedback incentive, each RM rejects reputation queries from agents that haven’t submitted any feedback for too long time.

CASE STUDIED

The system has been proved implementing and adopting the distributed HTML document retrieval system illustrated in [10]. Such multi-agent system allows users of a community to share documents by means of specialized software agents: Interface agents, Librarian agents and Neural Classifier agents.

Interface agents are used by end-users to specify document search constraints and to return matching documents. It uses the Neural Classifier agents available in the network, to perform the document searching/classification. Each Neural Classifier agent has an embedded neural network able to classify document belongs to a predefined set of topic classes.

Librarian agents publish the documents which their users have decided to make available in the network. Their task is to provide to Neural Classifier agents the unclassified set of holding documents.

For each search request, the Neural classifier clones itself in Librarian agents locations. Each clone classifies the set of documents provided by the Librarian agent, and sends the matching document to the Interface agent.

The experimental results regard the following 6 “Yahoo!” classes: Arts, Business, Computer, Government, Health, Recreation. System training has been performed using a random set of 1080 “Yahoo!” pre-classified HTML pages (180 per topic) and the performances have been tested with 2160 new documents, uniformly distributed over the above classes. The classification rate has been determined comparing the system results with the humans ones on similar situations. The system is very efficient: the experimental results show that in the best case a classification error of 10% is obtained [10].

The resulting system based on authorization of users allows making available documents only to trusted partners. Reputation, on the other side, allows users to discern better Neural Classifier (Librarian) agents reducing dramatically the risk of addressing to malicious or hopeless agents [13]. Besides, we have made use of 56 people, usually attending our laboratory, to test the authentication system easy-of-use. It is worthwhile to point out that 2 people refused to give us their fingerprints, while the remaining 54 people were willing to contribute. Experimental trials, conducted with 54 fingerprint pairs, show the feasibility of the system, which attained a 100% rate of success [9].

CONCLUSIONS

In this paper we have addressed some challenging problems of multi-agent systems: (i) safeguard of users (agent owners) against malicious users they could interact with; (ii) access restriction to protected resources.

The first issue has been tackled using authentication process based on biometric features and creating a reputation-based trust system. The latter issue has been tackled by means of an access control mechanism based on policy files. The proposed approach does not demand high system overhead and meets the scalability requirements of every real system. The use of biometric characteristics for ownership determination, will allow an increased level of security for all those applications, like e-commerce, e-banking, and so on, for which the determination of agent ownership is crucial.

To prove the effectiveness of the developed security infrastructure, implemented as an extension of the JADE-S platform, a distributed document retrieval system was implemented.

REFERENCES

DEVELOPING ANTICIPATORY AND AFFECTIVE COMPETENCES IN MULTI AGENT SYSTEMS

Cristiano Castelfranchi and Rino Falcone and Michele Piunti
ISTC - Institute of Cognitive Sciences and Technologies - CNR - National Research Council, Italy.
44, S. Martino della Battaglia - 00185 Roma - ITALY
c-mail {c.castelfranchi, r.falcone, michele.piunti}@istc.cnr.it

ABSTRACT

In this work we embed some anticipatory mechanisms in an agent architecture, modeling affective behaviours as effects of surprise. We look at the advantages of becoming cautious, wary and careful. Through experiments discussion we show how this approach increases not only agent opportunism and reactivity but also anticipatory capabilities for planning and reasoning.

INTRODUCTION

For over a decade the major claim about agent systems research was the necessity to deal with all those problems that are intractable using conventional approaches. Early generations of complex systems were designed for static worlds, assuming omniscience about the external environment, hence satisfying only a restricted domain of problems. In order to cope with real world applications we have to consider new classes of constraints like partial and uncertain knowledge, resource narrowness, social interactions and chaos (Bratman et a. 1988, Pollack et al., 1998). We assume that the relevance of this problem is not only for theoretical modeling but that will soon become foundational as in the virtual world and in the physical one, as for software cognitive agents and for autonomous robots. To have anticipatory competences, to experience affective behaviours (e.g. surprise), and to exploit the various functions of them will soon be crucial, not only for curiosity and exploration in unknown environment (Macedo and Cardoso 2001, 2004), but much more for learning and attention capabilities, for prudenty adjusting behaviours, intention reconsideration (Parsons et al. 2000), and for management of social relations like trust and suspicion (Castelfranchi and Falcone 1998), coordination and reliance etc.

HOW SURPRISE AFFECTS AGENT BEHAVIOUR

Among several definitions in literature, is that artificial surprise is due to -and is a signal of- a mismatch between the agent expectation (internal state) and the actual perceived input (perceived data) (C. Castelfranchi and E. Lorini, 2003). Its internal signal alerts the agent and can change its behaviour both with: Long term effects (becoming more accurate and less self confident in predictions; memorizing anticipatory signs of danger; memorizing different kinds of environment; learning more or less safe plans and actions for a given goal in a given condition; etc.); Short term effects, like redirecting attention, searching for additional information, or becoming prudent.

Without examining deeper surprise sources and dynamics, we here notice that in risky, unknown, harmful environments agent cautiousness can be a consequence of a kind of Surprise. Here we mainly model this effect of surprise and its possible advantages. Given this, it is possible firstly to characterize different kinds of agents, various stable 'personalities' (a very cautious agent vs. a rash agent, Schut et al. 2000), and to experiment their performance in more or less dangerous environment; secondly it is possible to build an adaptive agent able to run-time adjust its "degree" of prudence and carefulness to different environments or situations. We provide in this paper a first modeling of this kind of agents, and we show experimental results illustrating their different performances.

EXPERIMENT DESIGN

A test bed scenario is designed as simulation where entities and environment are represented as autonomous agents and artefacts. Environment is a 2D land map where sets of walls and gates (that can be open or closed) delimit rooms, corridors and areas where entities are able to move. Moreover environment presents the following object ontology: a set of special location seats symbolic reference points: we define them as location of interest (LOI) containing noticeable world objects: Repository, House, Tree; Two kinds of Food objects appear with modifiable frequencies, near to House and Tree. Foods rise at fixed location with a modifiable "reward" value; Frozen puddles, are located across rooms; Fires objects behave according to a two state cycle periodic function: in each period their first shape is a "Smoke" premonitory state, then they become real dangerous "Fire" objects. For each period, fires change their location with discrete movements. Environment holds delimited risky areas where rise of fires can be related with agent presence producing dangerous zones.

At an high level of abstraction, we consider agents as mobile entities with sensors, reasoning and effectors components and we characterize them with the following tuple of dynamic resources:

\[ \text{Ag} = \langle \text{En}, r, \text{Sr}, s \rangle \]  \hspace{1cm} (1)

\( \text{En} \) indicating the instant amount of energy, \( r \) the range of vision where sensors can retrieve data, \( \text{Sr} \) sensor sample rate, and \( s \) agent instant speed. We assume agents burn energy according to the combination of previous resource allocation (e.g. the more speed and sensor-rate is high, the more agent will spend energy). Moreover agents have a data set representing knowledge and internal state: it is handled as structured belief sets (e.g. object location, Environment map and knowledge about predefined paths between rooms and LOI, data about expectations, etc.).

In a Goal oriented fashion, first order objective for entities is to collect Food objects in the Repository to obtain rewards and recharge energy. We have suppose the task composed of the following recursive workflow of actions: 1) Look for Food with (supposed) best reward. Because of agents know each class of Food rises near a proper LOI, this introduces agent explicit quantitative expectations about the eventual Food presence and value. Each lookForFoodPlan is associated to a specific LOI and constantly monitored by a motivational level, determined merging both expectations on food reward and expectations on food presence near the LOI.
\[ \text{Motivation}_{\text{LOI}} = < \text{Exp}_{\text{Rew}}, \text{PF}_{\text{LOI}} > \]

\( \text{Exp}_{\text{Rew}} \) is determined averaging rewards for last delivered Foods, while \( \text{PF}_{\text{LOI}} \) indicates the probability to discover Foods near LOI: results of actions are used both to reinforce beliefs and associated expectation through a feedback. In turn, motivation for epistemic actions (EA) toward LOI is a subjective expected utility merging expectation values and EA results: agent meta-level-comparison reasons motivations and select for the specific plan toward best expected LOI, according to a -greedy policy. 2) Go to the identified Food location and pick up it. Identifying with sensors a set of Foods, agent add them to the belief base and head for the nearest one, observing topology constraints and obstacle bounds. 3) Transport Foods (one per time) from the original location to the repository and deposit them. Releasing Foods in the repository, agents obtain a reward calculated from the basis of the original food reward decreased with a decay factor straight depending on the transport time interval.

Along the presented workflow, agents can run up against Fires or Frozen Puddles: in these cases agents present a general short term reaction: actions and speed are obligated to reduce, furthermore a greater amount of energy has to be spent. Agents use a belief modules referring to a set of paths (defined as list of location to pass through) in order to routinize crossing rooms and LOI.

Starting from a traditional approach to the BDI systems, we developed an architecture in order to deal with 'the future', where intelligent agents build specific mental representations for explicit expectations (Lorini and Falcone 2005).

CAUTIOUS BEHAVIOUR

Handling explicit expectations is a necessary capability for agents that have to predict unexpected events and dynamics in the world. Expectations are involved in two phases of meta and means-end reasoning. As for the former, including expectations in the deliberation process elicit them to affect the reasoning cycle not only as a simple value, but as "first-class-object". Expectations are modelled to affect the traditional Reaction-Deliberation mechanism which traditionally relies on belief formulae (see Thangarajah et al. 2002; Braubach et al. 2005). As for the means-end reasoning, an ad hoc mismatch evaluator was built to deal with value discrepancies agents notice between expectations (what the agent foresee) and the real perceived data (e.g. quantify if Food reward obtained comply with corresponding reward expectation). Since expectations "have to be tested" continually with the world, a component was built to update them using feedback signals filtered from the sensors.

Caution is straight based on expectation: and we are interested to the anticipatory dimension of caution, as it is strictly related to a form of expectation: as agents uses and represent expectations to select best expected action from a set of available ones, as caution guides agent in the selection of less risky action between the ones in repertoire. Since we consider plans as workflow of self-contained actions, we associate to each workflow a risk-value depending on the amount of dangerous accidents registered in agent lifetime.

With the same principles used to realize components for expectations, we define a caution handler component. Expectation-driven agents act considering explicit subjective expected utilities in meta reasoning. Furthermore, during execution agents experience short term responses against unexpected events: a momentary lapse in the control of execution, mobilize their computational resources and increase energy consumption. Sensors and all the available epistemic resources are immediately allocated to the source of the event (e.g. focusing on unexpected objects and enhancing short-term sensor epistemic actions). Spending more resources for attentive processes has direct effects in reducing promptness and speedness and side effects in bodily reactions, as further energy consumption.

In order to compare different caution competences, we define a couple of agent with stable 'personalities' (with routinized intention deliberation). To capture environment regularities (e.g. dangerous areas) we enable watchful agents to chose plans with the expected lower risk-value (and accordingly the higher safety) while rash agents prefer to chose the quickest and simplest ones.

A more advanced class of cautious agents is designed with specific competences for adaptation of the caution level, in order to deal with environments having high dynamism. The caution properties can be reflected on the tuple of agent independent resources (1), saying that an agent increasing caution level and becoming watchful has to re-allocate resources a) Reducing speed s; b) Increasing Sensor rate Sr; c) Increasing range of vision r. These adaptations, heavily modify agent behaviour. Hence, some remarks has to be done. First these are independent dimensions and agents can follow modifyable policies for resource allocation to deal with sets of asynchronous events. Secondly, it should be pointed out the presence of a implicit trade-off: these responses elicit antithetical effects upon agent internal state. On the one side we identify negative effects: watchful agents have more energy consumption because of resources utilization. We suppose in effects b) and c) a higher energy consumption, partly paid by a lower use of engines for a).

Furthermore going slower constrains to catch up less Foods (especially in highly populated environment with bounded resources and competition). In these cases increasing attentive resources reduces the successful chances: the more the agents are quick, the more foods they will reach. Moreover being slow compels to suffer a bigger decay for Food reward at the 'drop action' in the Repository, because of the enlarged transport time. On the other side we suppose the benefits are in anticipating risks and watching faraway objects (looking ahead, e.g. enhancing r). Watchful agents are enabled to choose, between plans, the less risky pondering the memorized risk-value. Furthermore increasing attentive resources could imply increment of chances in the sense that with a bigger range of vision agents spend less resources for harm avoidance, and more ones for Goal purposed intentions. To deal with adaptive caution, we designed agents using anticipatory capabilities foreseeing and facing with coming negative events. An internal k-length buffer for registering events was constantly updated and evaluated: when the number of negative events exceed fixed threshold in a k-length time interval, the adaptive agent autonomously deliberates to increase the caution level. They move from one another on the basis of the negative events stored in their buffers during the last time interval and they shift among three kind of more or less cautious behaviours: default, watchful and rash. Adaptive agents adapt on the fly
caution configurations to anticipate the dangerous risks of the crossing area.

**EXPERIMENTAL TEST**

To measure agent effectiveness we considered environments with different levels of risk and processed statistics to monitor agent energy in function of simulated time (fig. 1).

![Figure 1. Agent average energy comparison in safe, risky and unsafe environment](image)

**Experiment 1.** Agents act in a safe world without risks. No Fires and no frozen puddles are present. Best mean performances are for the rash agent, because of his higher speed and no resource consumption for useless attention/caution capabilities (fig.1).

**Experiment 2.** Agents act in a risky environment with two Fires and two frozen puddles. In the same time interval, Adaptive agent outperforms the others: in these cases a stabilized cautious agent prevents accidents, while adaptive agent (with a modifiable caution) is able to minimize negative effects until its buffer reaches the threshold to be watchful. On the contrary, rash agent is unable to react in safety mode because of his fixed low level of Caution.

![Figure 2. Agent energy in unsafe environment](image)

**Experiment 3.** Agents in a unsafe environment with 4 Fires and 3 frozen puddles. It is interesting to note the different slopes of the functions (caused by the different energy consumption). Peaks indicates goal achievement while precipitous changes of slope indicate unexpected accidents (agent near Fire or near a Frozen puddle) (fig. 2). Best energy performances come from adaptive agent that is able to optimize damages and resource consumption. Energy trend for the rash agents falls with deep peaks. Mean performances of cautious and adaptive agents that are comparable (fig. 1); even if the adaptive agents continuously try to modify their behaviours during the simulation (adapting them to the environment) at the end their global performances result comparable with the ones of the watchful agent that in some sense is “built” for well performing in a highly unsafe world.

**CONCLUSIONS**

We have shown a first model of a cautious agent and an implementation of it through the BDI paradigm. For implementing and simulating caution in a real way we developed expectation-driven software agents. We also experimented cautious agents in a risky environment for analyzing their performances both on the basis of the environmental features and of the model of caution implemented. The experiments represent a first attempt to evaluate our model of caution. In future works we will develop a more accurate analysis of the relationships among the different parameters characterizing caution (speed, senscr rate, and range of vision) with respect to the risky properties of the different environments. We are also interested in implementing and exploiting the cognitive surprise and its functions. We consider these cognitive functions as a fundamental instrument for the anticipatory behaviour that is the real challenge for both the future cognitive software agents and autonomous robots.

**Acknowledgment:** This work is based on the EU research project "MindRACES" within the Cognitive Systems area. Contract Number: FP6-511931. See www.mindRaces.org.

**REFERENCES**


L. Mcedo and A. Cardoso, ‘Exploration of unknown environments with motivational agents’, In 3rd Int. Conf. on Autonomous Agents and Multi Agent System (AAMAS04), Columbia University, NYC, USA, (2004).


ToothAgent: Brushing on Your Behalf

Sameh Abdel-Naby and Paolo Giorgini
Department of Information and Communication technology
University of Trento
{sameh, paolo.giorgini} @ dt.unin.it

ABSTRACT
Advanced mobile services at-ease-invention seekers, or early adopters, and new-invention bystander, or late integrators depend on friendliness, reliability and smoothness of technology. The challenge is to combine Software Engineering with its Agent-Oriented programming Techniques while utilizing Multi-agent systems to encourage mobile users to use Advanced Mobile Services. In this paper we introduce a new form of the previously implemented architecture: ToothAgent; we perform a simple amplification process that helps connecting ToothAgent newly-proposed schemes, and demonstrate the integrations of the foreseen architecture.

INTRODUCTION
As long as it is people’s desires that drive software engineering R&D, at-ease inventions need to be added to ToothAgent-like systems, to cope with these desires. Nevertheless, previously introduced inventions still have some deficiencies. For example, in the 2005 MoPiDiG [1], best applied in organizations where users grouping and categorization are needed (such as museums), GROUP definition was not clearly maintained so it made the possibility to look at a single user as a group even if it is more feasible to merge other users of the same interests into a bigger single group. Also, the dynamicity of the ad hoc mobile network was considered generally without specifying different scenarios for different communication methods (e.g. logging in and out process duration through Bluetooth not as feasible as Wi-Fi and WiMAX). Thus, the role of AO community, particularly Multi-Agent Systems, should exceed facilitating current desires of people to foresee future services. A similar initiative was implemented by University of Trento MAS research group in collaboration with ARSLOGICA R&D lab, ITALY (for more info: www.sesaspa.it) [2] and we move now to explain and un and re Lego it further on.

ToothAgent
The ToothAgent [2] architecture is portal multi-agent-platforms-based that can be installed in server where Multi-agent techniques are enabled; each server handles a specific data service area, and system users can contact their personal agents using their Bluetooth integrated devices. Notably, ToothAgent is domain and technology-independent, so specific services provided by a server do not affect the system, and different technologies can be used in different sights.

This system allows users to express their interests and customize their accessible services according to location through basic interactions, categories and keywords (Figure 1). Application interactions are maintained through a user-friendly interface that requires the user to insert variable parameters. A transmission process between the mobile device and the system is executed after finishing the configuration process, and again re-executed whenever a proper server is allocated, this happens after the searching process offered by the system and within the mobile Bluetooth coverage range.

![Figure 1: Interactions of System’s Components](image)

As figure 1 shows, through an input/output relationship, ToothAgent user will be taking over 2 authorities: the PC part, where user will configure his ToothAgent-based application for requested services; and the mobile device part, which the user able to function through the Bluetooth harbor and retrieve relevant data from the PC (wireless relationship). At the same time, this PC will be communicating with ToothAgent services server throughout a wired link requiring list of servers’ locations and services places. Another pioneering feature in ToothAgent architecture is the ability of users to recover queued results throughout granting system clients the accessibility to check predefined demand results, regardless of whether the user is within the Bluetooth range of the server or not. PC operating software enables users to search and identify servers and services that are listed in the ToothAgent architecture database. Upon selecting a service, users will send relevant information, which facilitates the service customization process. The initial version of ToothAgent has been implemented and tested in the University of Trento.
taking the case of students’ interests in buying used books (http://www.dit.unitn.it/~pgiorgio/ToothAgent)

**ToothAgent: FORESEEN ADAPTATIONS**

**Enhancements**
ToothAgent has introduced ID-giving option based on Bluetooth MAC address. However, this is insufficient in case of integrating different communication methods as explained below. Users register in a system in specific time and date; a combination between both of them and the user family name’s first 3 characters (optional) will create an exclusive ID that recognizes its user’s profile. In addition, embedding an automatic logging-in technique into the ToothAgent mobile-based application, and establishing a connection within the available network, will avoid relying on Bluetooth unique ID and will make it possible to lead to effective results.

**Offline messaging capabilities integrated with Time-to-live (TTL) data packet tags,** by allowing clients to select a main server to be acting like their service main faucet. If a user is out of the Bluetooth coverage area, implementing a function that replaces Bluetooth messaging with mobile (SMS) will be feasible. In case the user is still out of mobile network coverage, then a message should be initiated and saved in what we call the Faucet -Service-Server and then sent to the user whenever he is available in the server communication range. Otherwise the message should be killed according to its TTL tag. Faucet Service Server can be selected by location (e.g., the nearest one to user home or office address), or by checking the visited servers list (already implemented in ToothAgent) and assign the most visited one to the user ID.

**Enabling different communication types and models** will increase the smoothness of service, a user can be out of the system recognized communications range, yet he is using his PC for work or connected to the internet. So enabling the PC ToothAgent application to recognize its owner availability and converting the structure workflow into a chatting (or similar) service will make a user’s access goes to more realistic edges, especially if other users are using their PCs as well (e.g. in a university or working place). System should not wait for users to be available, but search for them. A value-added example could be auction-style service fulfilling gaps between buyer and seller (e.g., used books).

**Preferred Meeting Spots Definition.** If 2 users agreed to take a specific action (e.g. buyer and seller to meet and finalize the deal), a call or any external communication method should be taken in order to define a meeting point. If the system database is developed in a way that recognizes the users’ locations, it will suggest a meeting point according to pre-defined spots listed in the database, and send this suggestion along with the initial messages (e.g. this book costs $20 the meeting point is at the reception). Accordingly, extra communications hassles will be avoided, linking people on different scales and locations will be feasible, and taking decisions on the client’s side will be faster.

**Reputation Handling.** the system can implement a grouping action, in which categorizing people will be done according to their interests and needs or working fields and locations. The fundamental concern is to have users in groups, but acting and exchanging requests individually, so the maintainability of system group reputation will be achievable, and controlling participants behaviors will be monitored by all of the group members. This will help the service improvement and security. If a user is informed that he is linked to a specific group (e.g., friends) and any malicious transaction will affect the entire group reputation, the system will not have false operations and the traffic utilization will be purified.

**Matching interests with similarities,** similar interests recommendation system is now common among major online eCommerce portals and service providers. For example, if one is looking for *Thinking in Java* book, and someone else is offering *Thinking in Java Live Presentation* of the book’s author in a videocassette, matching this two requests as a system suggestion will increase the number of successful transactions.

**INTEGRATIONS**
The integration between ToothAgent mobile based application and the existing advanced mobile services platforms and portals (e.g., WAP) will enlarge the variety of system options and facilitate ToothAgent services accessibility. For example, if a user has suddenly noticed that he/she is getting many requests with high prices in reference to her *The Chamber* book selling request, that might lead the user to check the book availability online so it affects her proposed price. As long as ToothAgent connected to the updated products-list, this will function as a measurement balancer among system users, and gives advises accordingly.

**EXPECTATIONS**
We expect ToothAgent, after enhancing its connectivity and adding to it extra features and transactions conditions, along with its mobile-based application modifications, to be able to cover larger areas, greater amount of transactions, several services types and, finally, a greater number of users.

**TECHNOLGY MERGER**

**Bluetooth**
Using Bluetooth to operate messaging service within a Multi-Agent oriented architecture will cause a serious reliability problem. Whenever Bluetooth enabled users enter into a certain communication ring, with people walk in a shopping mall for example, repeated logging
and shutting for the connection is required since the coverage area is limited. Also, Bluetooth radios operate in the unlicensed 2.4-GHz band that that requires a large number of unrelated devices which would eventually interfere with one another and, as a result, for devices to achieve successful transmission, they have to wait; a network slowdown.

**Wi-Fi and WiMAX**

Wi-Fi - *Wireless Fidelity* solutions presented in [3] are reliable in handling Agent-oriented-based messaging platforms. The combination of large-area coverage with communication speed explains the widespread use of Wi-Fi applications. It uses IEEE 802.11 standard and it has got 2 common approaches: Directional Antenna-based and Mesh-network topology based. Once the security problems of these technologies are recovered, it is recommended this integration in any Agent-Oriented application. Worldwide Interoperability for Microwave access WiMAX presented in [4] covers up to 50-Kilometers of service area with high data transmission rate, up to 280 Mbps/station. It is highly likely that WiMAX would replace Cellular networks, which would affect any mobile-related services.

**BUSINESS AWARENESS**

Agents negotiate with one another to gain access to other resources and possibilities even though a business agent is an autonomous entity [5]. A common problem is for agents to directly communicate business aspects and then merge them into a meaningful application [5] and that what led agents reach the self-representation abilities within a business structure.

M. P. Papazoglou [5] has defined a multi-agent e-business environment through allocating agents into a different layers depending on the business process requirement. The four basic forms of business agents can be summarized as follows:

**Application agents**: functioning vertically and individually according to their experiences in a single area within a business domain (e.g. e-commerce); they supply other agents in the network by supplying them with information accessibility and knowledge distribution sources. ToothAgent can achieve this function. For example, if two buying agents sending requests at a time to a selling agent with different prices, but both within the pre-defined average price, the selling agent should be able select and communicate the higher price to the user.

**Personal agents**: working directly with users to support the agent-based business architecture with the presentation, organization and management of users’ profile, other agents requests and information collection. ToothAgent is not applicable to this level, since no application monitoring system is implemented within the configuration process through the PC-based application.

**General Business Activity Agents**: operating certain processes that enable network navigation until information and service requests are achieved. ToothAgent is applicable to this level as the multi-agent architecture it uses is mainly concerned with finding trade partners and negotiating similar interests and needs.

**Information Brokering Agents**: determining information path contained by a network or solving ordinary problems. Brokering agents are capable to preserve, enter and update distributed directory services [5]. ToothAgent is also applicable to this level, since agents which are implemented in this architecture have the possibility to access the users network searching for demanding information, sort it, and finally return it to the system users with expecting results.

**CONCLUSIONS**

This paper embarked succinctly on the motivation for implementing Multi-agent based architectures, and the limitations for the current ones, and then turned to discussing ToothAgent and the emerging communication technologies with ToothAgent-like systems and made some recommendations of the standard use of each. After we allocated ToothAgent within a pre-defined Agent based business architecture, we came up with a conclusion that ToothAgent has missed the personal agents business implementation concept and was acceptable from the application agents perspective. Accordingly, we proposed some novel ideas on the improvement of these architectures and our expectations after the implementation of these ideas. To empirically test our ideas, we plan to test integration of ToothAgent-based architecture first in the University of Trento Wi-Fi network, if successful, in a larger coverage area.

**REFERENCES**


AGENT RELATED TECHNOLOGIES
CHOOSING THE RIGHT DESIGN PATTERN: THE IMPLICIT CULTURE APPROACH

Aliaksandr Birukou, Enrico Blanzieri, Paolo Giorgini
Department of Information and Communication Technology
University of Trento, via Sommarive 14, 38050 Povo (Trento), Italy
e-mail: {aliaksandr.birukou, enrico.blanzieri, paolo.giorgini}@dit.unitn.it

ABSTRACT

Design patterns represent reusable solutions to problems that have been proved to be useful in different contexts. An experienced programmer can choose a suitable pattern for a given problem effectively. However, for an inexperienced programmer this is a very hard task. We propose a multi-agent system that supports programmers in choosing the design pattern suitable for a given problem. Personal agents in our system produce knowledge transfer among users, allowing for the reuse of experience in choosing design patterns.

INTRODUCTION

“With more than 20 design patterns in the catalog to choose from, it might be hard to find the one that addresses a particular design problem, especially if the catalog is new and unfamiliar to you.” (Gamma et al., 1995). This shows that selecting a design pattern was a problem even ten years ago. The continuously growing number of patterns (e.g., W. F. Tichy’s catalog at http://wwwipd.ira.uka.de/tichy/patterns/overview.html contains over 100 patterns) complicates the problem even more. For an inexperienced programmer it is very hard to choose the right design pattern and tools assisting in this process become of the utmost importance.

Unfortunately, there is still a lack of systems that guide a programmer in the selection of design patterns (see Kung et al., 2003; Gomes et al., 2003) for examples of such systems). Most of the current approaches dealing with patterns suppose that it is the programmer who makes the choice of the pattern (Ó Cinnéide and Nixon, 2001).

This paper addresses the problem of design pattern selection and proposes an approach that considers the problem from a social point of view. We propose the use of a multi-agent system based on the Implicit Culture framework to help programmers in selecting patterns by getting suggestions from the group. In the system, the problem faced by the programmer is compared with those faced previously by colleagues and suggestions about the most suitable design patterns are provided.

IMPLICIT CULTURE FOR DESIGN PATTERN SELECTION

This section presents an overview of Implicit Culture (IC) and Systems for Implicit Culture Support (SICS) (see Blanzieri et al., 2001) for more details. It also contains the formalization of the problem of design pattern selection.

“Only experienced software engineers who have a deep knowledge of patterns can use them effectively. These developers can recognize generic situations where a pattern can be applied. Inexperienced programmers, even if they have read the pattern books, will always find it hard to decide whether they can reuse pattern or need to develop a special-purpose solution.” (Sommerville, 2004). The difference between two programmers is that the experienced programmer uses implicit knowledge (in particular, his experience) about the problem. Knowledge is called implicit when it is embodied in the capabilities and the abilities of the community members. It is explicit when it is possible to describe and share it through documents and/or information bases. To select suitable design patterns the inexperienced programmer should acquire the implicit knowledge of more experienced programmers.

We argue that it is possible to shift the behavior of inexperienced programmers in design pattern selection towards the behavior of experienced programmers by means of suggesting patterns which are more suitable for the current design task. To determine appropriate patterns we use the history of previous interactions with the system, i.e., what patterns have been chosen for similar design tasks. We call the behavior of experienced programmers related to the pattern selection a community culture. When inexperienced programmers start behaving similarly to the community culture the knowledge transfer (performed by the SICS) occurs. The relation characterized by this knowledge transfer is called “Implicit Culture” (Blanzieri et al., 2001).

For example, let us consider a programmer that needs to define an interface for creating an object, but let subclasses decide which class to instantiate. Let us suppose that for an experienced programmer it is obvious to use Factory Method pattern here. If the system is able to use previous history to suggest that the novice uses Fac-
tory Method pattern and he actually uses it, then we say that he behaves in accordance with the community culture and the IC relation is established.

The general architecture of a SICS consists of the following three components: the observer, which uses a database of observations to store information about actions performed by the user; the inductive module, which analyzes the stored observations and applies learning techniques (namely, Data Mining or Machine Learning) to develop a theory about actions performed in different situations; the composer, which exploits the information from the observer and the inductive module to suggest actions in a given situation.

In terms of our problem domain, the observer saves information about the problem, which patterns have been proposed as a solution and which pattern has been finally chosen. The inductive module discovers problem-solution pairs by analyzing the history of users’ interaction with the system. A set of problem-solution pairs form a theory. The goal of the composer is to compare a problem faced by the programmer with the problem part of the theory and to suggest the corresponding solution part. If this step fails, the composer tries to match the problem with the solution by calculating the similarity between the descriptions of the problem and the patterns.

To formalize the problem of pattern selection for a community of programmers we need to decide how to describe patterns and design problems and how to match a given design problem with an appropriate pattern.

According to (Gamma et al., 1995) a design pattern has the following four parts: the name, the problem, the solution, and the consequences. The most interesting part for us is the problem, since it contains the description of the task solved by the pattern. For instance, in the catalogue of GoF patterns, the information about the problem is contained in the section “Intent”. The system should match this description with a description of the design problem. We assume there exists a (semi)formal or textual description of the problem faced by the programmer and of the problem solved by the pattern. In IC terms these problems constitute situations. Each situation is associated with actions performed by agents (programmers) on objects. The actions can also have attributes, which are features that can be useful for the analysis. In our application, the SICS analyzes the actions presented in Table 1 (see (Birukou et al., 2006a) for more details). Every action is recorded as being performed in a context of a certain project, characterized by an optional attribute project_name. We introduce this attribute, because a project that has some specific requirements can influence the choice of patterns. The goal of the SICS is to offer the programmer actions which have been performed in similar situations, i.e. to suggest the use of patterns that have been chosen for solving similar problems.

<table>
<thead>
<tr>
<th>action</th>
<th>objects</th>
<th>attributes</th>
</tr>
</thead>
<tbody>
<tr>
<td>request</td>
<td>problem, description</td>
<td>project_name</td>
</tr>
<tr>
<td>apply</td>
<td>pattern, problem, description</td>
<td>project_name</td>
</tr>
<tr>
<td>reject</td>
<td>pattern, problem, description</td>
<td>project_name</td>
</tr>
</tbody>
</table>

Figure 1: Architecture of IC Multi-Agent Platform.

**USING IMPLICIT CULTURE MULTI-AGENT PLATFORM FOR PATTERN SELECTION**

In this section we briefly describe the architecture of the multi-agent system for design pattern selection. We start from the description of a general IC Multi-Agent platform which was previously applied to the problem of web link recommendation for a community of users (Birukou et al., 2005) and to scientific publications search (Birukou et al., 2006b). The platform is implemented using the Java Agent DEvelopment framework (JADE, http://jade.tilab.com/). The general architecture is depicted in Figure 1 and it consists of the following components (see the JADE documentation or (Birukou et al., 2006a) for more details). A user submits requests to the system using the interface on the client side. A request contains the description of the design problem. A Personal Agent (PA) is a software agent which assists the user in choosing a suitable pattern. It uses dedicated API to query several information sources, which contain information about design patterns. The Directory Facilitator (DF) provides agents with the IDs of other PAs and uses the SICS module to produce recommendations about patterns that have been selected for similar problems. The recommendations are based on previous experience of the users.

A typical usage scenario is as follows: a user submits a request, expressing a design need. The personal agent contacts the DF. The DF stores the request action in the database of observations and uses the SICS module to find patterns chosen for similar problems previously. The suggestions are sent to the personal agents and are shown to the user. Moreover, it is also possible to show
descriptions of similar problems discovered, because this can help the user to take a decision. Finally, being able to see a list of similar problems faced by his colleagues, the programmer could have a short talk with them to share their experience.

EVALUATION AND DISCUSSION

We are currently working on the implementation of information sources. We use Apache Lucene (http://lucene.apache.org/), an open source search engine library, to create an index of the pattern repository and to search for pattern descriptions using this index. So far we have built a repository of 23 design patterns from (Gamma et al., 1995). Personal agents use API to access Lucene searching capabilities. We have not performed the system evaluation with real users yet. However, in (Birukov et al., 2005) we presented numerical results obtained using the simulator developed for the application of the IC Multi-Agent Platform to web search. The results have shown that an increase in the number of users causes an increase in the recall of the suggestions produced by the system. We think that the problem of design pattern selection is much related to the problem of selecting web links relevant to keywords and that our approach will be also useful here.

The Expert System for Suggesting Design Patterns (ESSDP) (Kung et al., 2003) and REBUILDER (Gomes et al., 2003), similarly to our system, suggest design patterns to solve problems faced by designers. There are several differences between our approach and ESSDP. Firstly, ESSDP assumes knowledge acquisition (human experts must fill in the knowledge base) as the primary step. Differently, in our system the SICS learns from the interactions with users (both expert and novices), without any initial knowledge base, allowing for continuous improvement of suggestions. Secondly, our architecture is not restricted to the use of a rule-based knowledge base assuming that different learning techniques can be adopted. REBUILDER uses Case-Based Reasoning (CBR) techniques for the pattern selection. The main differences between IC and CBR are the following: while CBR solves a new problem by remembering a previous similar situation and by reusing information and knowledge of that situation, IC helps to solve a problem but does not produce the solution directly; in CBR, the problem is represented explicitly (the case), while IC does not deal with an explicit representation of the problem, only with the implicit information about it. The system can be extended in a number of ways. For instance, it might also be applied to the selection of the most appropriate version of the pattern, when several versions are available. A semiformal representation of the design problem (e.g., class or activity) or a formal framework, such as the non-functional requirements (NFR) framework (Gross and Yu, 2001) can be adopted for the description of design problems/design patterns.

We also think that the problem we address is related to the problem of web service composition, where to solve a sub-problem it is necessary to find a web service which suits some design need (Lazovic et al., 2006). Our approach can be also used in tools for refactoring the old code using patterns, e.g., (Ó’Cinnéide and Nixon, 2001), at the stage of selecting the pattern: a programmer can be provided with suggestions about patterns used in similar situations previously.

CONCLUSION

A system that helps programmers in choosing design patterns suitable for a given task has been described. The system takes into account the social part of the problem, providing users with suggestions from other community members. As future work we would like to conduct several experiments with real users.

ACKNOWLEDGEMENTS

This work is funded by EU SERENITY, COFIN, MOSTRO and QUIEW. We would like to thank the anonymous reviewers for their helpful comments.

REFERENCES


Gamara, E., R. Helm, R. Johnson, and J. Vlissides. 1995. Design patterns. Addison-Wesley.


KNIME: THE KONSTANZ INFORMATION MINER

Michael R. Berthold, Nicolas Cebron, Fabian Dill, Giuseppe Di Fatta, Thomas R. Gabriel, Florian Georg, Thorsten Meinl, Peter Ohl, Christoph Sieb and Bernd Wiswedel
Konstanz University, Department of Computer and Information Science
Fach M712, 78457 Konstanz, Germany
E-mail: Michael.Berthold@uni-konstanz.de

ABSTRACT

The Konstanz Information Miner is a modular environment that enables easy visual assembly and interactive execution of a data pipeline. It is designed as a teaching, research and collaboration platform, which enables easy integration of new algorithms, data manipulation or visualization methods as new modules or nodes. In this paper we describe some of the design aspects of the underlying architecture and briefly sketch how new nodes can be incorporated.

OVERVIEW

Large volumes of data are often generated during simulations and the need for modular data analysis environments has increased dramatically over the past years. In order to make use of the vast variety of data analysis methods around, it is essential that such an environment is easy and intuitive to use, allows for quick and interactive changes to the analysis and enables the user to visually explore the results. To meet these challenges a data pipelining environment is an appropriate model. It allows the user to visually assemble and adapt the analysis flow from standardized building blocks, at the same time offering an intuitive, graphical way to document what has been done.

Knime, the Konstanz Information Miner provides such an environment. Figure 1 shows a screenshot of an example analysis flow. In the center, a flow is reading in data from three sources and processes it in several, also parallel analysis flows, consisting of preprocessing, modeling, and visualization nodes. On the left a repository of nodes is shown. From this large variety of nodes, one can select data sources, data preprocessing steps, model building algorithms, visualization techniques as well as model I/O tools and drag them onto the workbench where they can be connected to other nodes. The ability to have all views interact graphically creates a powerful environment to explore the data sets at hand.

Knime is written in Java and it’s graphical workflow editor is implemented as an Eclipse (Eclipse Foundation 2005) plug-in. It is easy to extend through an open API and a data abstraction framework, which allows for new nodes to be quickly added in a well-defined way.

In this paper we will describe some of the internals of Knime in more detail. More information as well as downloads can be found at http://www.knime.org.

ARCHITECTURE

The architecture of Knime was designed with three main principles in mind:

- visual, interactive framework: data flows should be combined by simple drag&drop from a variety of processing units. Customized applications can be modelled through individual data pipelines.
- modularity: Processing units and data containers should not depend on each other in order to enable easy distribution of computation and allow for independent development of different algorithms. Data Types are encapsulated, that is, no types are predefined, new types can easily be added bringing along type specific renderers and comparators. New types can be declared compatible to existing types.
- easy expandability: It should be easy to add new processing nodes, or views and distribute them through a simple plug&play principle without the need for complicated install/deinstall procedures.

In order to achieve this, a data analysis process consists of a pipeline of nodes, connected by edges that transport either data or models. Each node processes the arriving data and/or model(s) and produces results on its outputs. Figure 2 schematically illustrates this process. The type of processing ranges from simple data operations such as filtering or merging to more complex statistical functions, such as computations of mean, standard deviation or linear regression coefficients to computation intensive data modeling operators (clustering, decision trees, neural networks, to name just a few). In addition most of the modeling nodes allow to interactively explore their results through accompanying views. In the following we will briefly describe the underlying
schemata of data, node, workflow management and how the interactive views communicate.

Data Structures

All data flowing between nodes is wrapped within a class called **DataTable** which holds meta-information concerning the type of its columns and the actual data. The data can be accessed by iterating over instances of **DataRow**. Each row contains a unique identifier (or primary key) and a specific number of **DataCell** objects which hold the actual data. The reason to avoid access by Row ID or index is scalability, that is, the desire to be able to process large amounts of data and therefore not be forced to keep all of the rows in memory for fast, random access. Figure 3 shows a diagram of the main underlying data structure.

Nodes

Nodes in Knime are the most general processing unit and usually resemble one visual node in the workflow. The class **Node** wraps all functionality and makes use of user defined implementations of a **NodeModel**, possibly a **NodeDialog**, and one or more **NodeView** instances if appropriate. Neither dialog nor view must be implemented if no user settings or views are needed. This schema follows the well-known Model-View-Controller design pattern. In addition, for the input and output connections, each node has a number of **Import** and **Output** instances which can either transport data or model(s). Figure 4 shows a diagram of this structure.

Workflow Management

Workflows in Knime are essentially graphs connecting nodes, or more formally, a direct acyclic graph (DAG). The **WorkflowManager** allows to insert new nodes and to add directed edges (connections) between two nodes. It also keeps track of the status of nodes (configured, executed, ...) and returns, on demand, a pool of executable nodes. This way the surrounding framework can freely distribute the workload among a couple of parallel threads or – in the future – even a distributed cluster of servers. Thanks to the underlying graph structure, the workflow manager is able to determine all nodes required to be executed along the paths leading to the node the user actually wants to execute.
Views and Interactive Brushing

Each Node can have an arbitrary number of views associated with it. Through receiving events from a HiLiteHandler (and sending events to it) it is possible to mark (the so-called HiLiting) selected points in such a view to enable visual brushing. Views can range from simple table views to more complex views on the underlying data or the generated model.

REPOSITORY

Knime already offers a large variety of nodes, among them are nodes for various types of data I/O, manipulation, and transformation, as well as data mining and machine learning, and visualization components:

- data I/O: generic file reader, ARFF and Hitlist file reader, database connector, CSV, Hitlist and ARFF writer.
- data manipulation: row and column filtering, data partitioning and sampling, random shuffling or sorting, data joiner and merger,
- data transformation: missing value replacer, matrix transposer, binnings, nominal value generators
- mining algorithms: clustering (k-means, sota, fuzzy c-means), decision tree, (fuzzy) rule induction, regression, subgroup and association rule mining.
- machine learning: neural networks (RBF and MLP), support vector machines\(^*\), bayes networks and bayes classifier\(^*\)
- statistics: via integrated R\(^*\)
- visualization: scatter plot, histogram, parallel coordinates, multidimensional scaling, rule plotters, line and pie charts\(^*\)
- misc: scripting nodes.

\(^*\): via external libraries or tools.

![Diagram of Data Structure and Main Classes](image)

Figure 3: A Class Diagram of the Data Structure and the Main Classes it relies on.

![Diagram of Node and Main Classes](image)

Figure 4: A Class Diagram of the Node and the Main Classes it relies on.

EXTENDING KNIME

Knime already includes new plug-ins to incorporate existing data analysis tools, such as Weka (Ian H. Witten and Eibe Frank 2005), the statistical toolkit R (R Development Core Team 2005), and JFreeChart (David Gilbert 2005). It is usually straightforward to create wrappers for external tools without having to modify these executables themselves. Adding new nodes to Knime, also for native new operations, is easy. For this, one needs to extend three abstract classes:

- **NodeModel**: this class is responsible for the main computations. It requires to overwrite three main methods: `configure()`, `execute()`, and `reset()`. The first takes the meta information of the input tables and creates the definition of the output speci-
fication. The `execute`-function performs the actual creation of the output data or models, and `reset` discards all intermediate results.

- **NodeDialog**: this class is used to specify the dialog that enables the user to adjust individual settings that affect the node’s execution. A standardized set of `DefaultDialogComponent` objects allows to very quickly create dialogs where only a few standard settings are needed.

- **NodeView**: this class can be overwritten multiple times to allow for different views onto the underlying model. Each view is automatically registered with a `HiLiteHandler` which sends events when other views have hilit points and allows to launch events in case inside this view points have been hilit.

In addition to the three model, dialog, and view classes the programmer also needs to provide a `NodeFactory`, creating new instances. The factory also provides names and other details such as the number of available views or a flag indicating absence or presence of a dialog. A wizard integrated in the Eclipse-based development environment allows to quickly generate all required class bodies for a new node.

**WORK IN PROGRESS**

Knime is continuously extended. A few extensions currently being actively under development are described below.

**Meta Nodes**

The ability to wrap a certain workflow into an encapsulating node provides a powerful abstraction mechanism. A sub-workflow can be included as a single component of another workflow, namely a meta-node. Such nested workflows introduce modularity and allow the user to design complex workflows while focusing on different level of details (abstraction). A meta-node can be exported to other users as a predefined module and allow to create wrappers for repeated execution as needed in cases such as, e.g. cross-validation, bagging and boosting, ensemble learning etc.

**High Performance Distributed and Parallel Computing**

Due to the modular architecture it is easy to designate specific nodes to be run on separate machines. The meta-node abstraction provides a mechanism to encapsulate workflows and to assign them to dedicated sever for distributed processing, resulting in a significant acceleration of the workflow execution. But to accommodate the increasing availability of multi-core machines, also the support for shared memory parallelism becomes increasingly important. Knime will offer a unified framework to parallelize data-parallel operations as well as the distribution of operations on a cluster or a GRID.

**Chem- and Bioinformatics**

A number of current projects focus on applications in the Life Sciences. Nodes to process gene expression data and high throughput, high content cell assay images are under development.

**Webservices**

Experimental nodes to access webservices via SOAP have been devised to call computation of chemical properties. Knime itself can also be seen as a potential server for a webservice itself, allowing external users to run predefined workflows.

**ACKNOWLEDGEMENTS**

We would like to thank numerous students of Konstanz University for continuous feedback and bug reporting. We thank, in particular, Kilian Thiel and Simona Pintilie for their work, respectively, on Sota and the Parallel Coordinates display.

**REFERENCES**


AN INTEGRATED DECISION SUPPORT SYSTEM BASED ON SOFTWARE AGENTS

Magid Nikraz and Parisa A. Bahri
School of Engineering Science and Parker Centre
Murdoch University
Dixon Road, Rockingham,
Western Australia, 6168
E-mail: m.nikraz@murdoch.edu.au

KEYWORDS
Software agent, integration, operational task, chemical plant.

ABSTRACT
A methodology for modelling of multi-agent systems using the JADE platform has recently been proposed. This methodology is applied to model the integration of operational tasks in a chemical plant. Initially, the operational tasks are outlined, then some insight is given into what integration entails in this context, followed by a discussion of previous work in this area. Next, a brief summary of the methodology recently proposed for the modelling of multi-agent systems using the JADE platform is given. Finally, it is shown how an agent-based approach is seen as a very effective modelling tool through application of the methodology to the pilot plant representing the Bayer Process at Murdoch University.

INTRODUCTION
Agent-based software engineering techniques are potential tools that can be used to model the integration of operational tasks in a chemical plant (Nikraz and Bahri 2005) to form an integrated decision support system. The common operational tasks in a chemical plant can be loosely classified as data acquisition, regulatory control, monitoring, data reconciliation, fault detection and diagnosis, supervisory control, scheduling, and planning. Integration in this context refers to the process of bringing these separate tasks together under a coordinated framework (Reklaitis and Koppel 1996). Though integration is highly desirable, its achievement is made very difficult by the complexity of today’s chemical plants and the varied solution methods employed for each of the operational tasks.

Agent-based software engineering techniques are suited for modelling the integration of operational tasks in chemical plants primarily because they provide the designer with a suitable mindset. The use of software agents provides a uniform means of communication between the modules representing the individual operational tasks, in addition to providing a way to encapsulate heterogeneous legacy systems and assimilate them into the integrated framework. The agent paradigm adopted by the methodology and hence this study means the modules realising the operational tasks can be implemented using whatever solution method desired, and distributed over several computers which may use different operating systems. Thus, the approach allows true flexibility. In addition, adaptability is another desirable characteristic of the agent paradigm, since new components can be added to the system with ease even after the system has been designed and implemented.

A methodology focusing on the popular FIPA-compliant (FIPA is the abbreviation for the Foundation for Intelligent and Physical Agents) JADE (Java Agent Development Framework) platform has recently been developed (Nikraz et al. 2006), which, in addition to covering standard considerations for modelling of multi-agent systems, takes into account several pertinent issues overlooked by currently existing methodologies (issues such as support for legacy systems and people). The methodology covers analysis and design (the design exclusively focuses on the JADE platform), and assumes that an agent-based solution has been chosen as the best alternative after consulting the literature (i.e. planning is not formally addressed). Like planning, implementation is not formally covered, but some details are provided on how to proceed with this phase.

In the following sections, the operational tasks and the concept of integration are defined, in addition to a discussion of previous work in this area. Then, a very brief summary of the methodology is presented, which forms the basis for the pilot plant implementation. In discussing the pilot plant implementation, it is explained why an agent-based solution is a relevant choice for modelling of the integration of operational tasks in chemical plants.

INTEGRATION OF OPERATIONAL TASKS IN A CHEMICAL PLANT
This section gives details of the common operational tasks in a chemical plant and then defines what is meant by the term integration in this context and previous work in this area.

The Operational Tasks
The operational tasks are traditionally depicted using a pyramidal as shown in Figure 1. Data Acquisition is the most basic component. Process data is acquired from the plant through sensors. The data is then used as a foundation for every other task. Regulatory Control occurs when the control system functions to counteract the effect of disturbances in order to maintain the output at its desired set-point. Monitoring detects the status of the system, i.e. if the process...
is at steady-state. Other functions such as sensor limit checking and exponential filtering of data may also be performed. Data Reconciliation is required because on-line process measurements are corrupted by unavoidable errors (caused by power supply fluctuations, signal conversion noise, etc.) during measurement and transmission of data. Two types of errors are usually present: random and gross errors. Random errors are small and are due to normal process fluctuations. Gross errors, on the other hand, are large and are due to incorrect calibration or malfunction of instruments, process leaks, etc. Before plant data can be accurately used, it is thus necessary to reconcile this data into meaningful values, and this is the objective of the data reconciliation module. Fault detection and diagnosis involves the tracking of process execution, detection of departures from normal operation (fault detection), and identification of cause (fault diagnosis). Faults include gross parameter changes in a model, structural changes, and malfunctioning sensors. Supervisory control involves the online calculation of set-points allowing the unit or plant to achieve best performance (based on economic models of the plant), while satisfying operational constraints. Scheduling is the determination of the timing and sequence in the execution of manufacturing tasks so as to achieve production targets in an optimal fashion. The time frame in this case is usually days to weeks. Planning is the allocation of production resources and assignment of production targets over a suitable time scale. The time scale for planning is long term, typically in the range of weeks, months or years.

Figure 1: Common Operational Tasks in a Chemical Plant

Definition of Integration

The term “integration” in this context refers to an automation architecture that unifies the various levels of the hierarchy defined in the pyramid model (see Figure 1). Integration can thus be seen as a goal which results from allowing the various levels of the hierarchy to interoperate.

In the past, much attention has been directed to integration of operational tasks from a manufacturing perspective, as made evident by the formation of the field known as Computer Integrated Manufacturing (CIM). The goal of CIM was and still is to produce a unified control and planning system, based on computer technologies, that will implement the full pyramid as a logically integrated whole (since monolithic implementations are obviously impractical). This will make the manufacturing part of the business more agile, allowing high-level planning to be automatically propagated to the plant floor.

While a great deal of work has been carried out on integration from a manufacturing perspective, very little attention has been directed to integration from the point of view of the chemical process industries (see “Previous Works” section below). Though there are some similarities between manufacturing and chemical processing, there are also some fundamental differences, mainly arising from the different nature of the operational tasks, which is why integration of chemical processes is defined as a separate problem. Nevertheless, in terms of integration, the end goal is the same, i.e., creation of an automation architecture that unifies the various levels of the hierarchy defined in the pyramid model.

Since the operational tasks defined above are often carried out by several different software programs, integration becomes an issue of interoperability between different software programs. These software programs which realise the operational tasks act separately, but may have common goals and rely on each other for decision making.

Previous Works

As already mentioned, very little research has been carried out which directly focuses on modelling the integration of operational tasks in chemical plants. After a review of the literature, it has been ascertained that there have been three major works carried out to date: A Framework for Integrated Process Supervision (Rengaswamy 1995), the CHEM Project (CHEM 2004), and the Coordinated Knowledge Management Method (Power 2004).

The first work by Rengaswamy is purely of an exploratory nature with no modelling tools being proposed. The CHEM project focused on the G2 expert-system programming environment (Gensym 2006) with additional use of certain middleware. The most recent work by Power used Petri-nets as the modelling tool. Power demonstrated that while Petri-nets are suitable for small plants, their use in large plants becomes difficult as the Petri-net will become too large and complex to manage.

For a more detailed review, the reader is referred elsewhere (Nikraz and Bahri 2006). It is clearly evident from this discussion, however, that agents have not yet been investigated as a potentially suitable tool, even though they are naturally suited to this task as discussed below.

METHODOLOGY SUMMARY

This section gives a summary of the methodology for the analysis and design of multi-agent systems using JADE. Those interested in further details are strongly advised to consult Nikraz et al. (2006). Before summarising the phases in the methodology, the definition of an agent assumed by the methodology is given.
What is an Agent?

The term agent is very broad and has different meanings to different people. However, on close observation of the literature, it is sufficient to say that two usages of the term agent can be identified: the weak notion of agency and the stronger notion of agency (Wooldridge and Jennings 1995). The weak notion of agency constitutes the bare minimum that most researchers agree on, while the stronger notion of agency is more controversial and a subject of active research.

The weak notion of agency denotes a software-based computer system with the following four properties (Wooldridge and Jennings 1995): (1) Autonomy: agents operate without the direct intervention of humans or others, and have some kind of control over their actions and internal state; (2) Social ability: agents interact with other agents (and possibly humans) via some kind of agent communication language; (3) Reactivity: agents perceive their environment and respond in a timely fashion to changes occurring therein; and (4) Pro-activeness: in addition to acting in response to their environment, agents are able to exhibit goal-directed behaviour by taking the initiative. The strong notion of agency is an extension of the weaker notion, and advocates additional humanistic, mental properties such as belief, desire, and intention (Shoham 1993).

The definition of an agent assumed by the proposed methodology is based on the weak notion of agency. Specifically, the methodology assumes the following definition for an agent: agents reside on a platform that, consistent with the weak notion of agency (i.e. an agent is autonomous, communicative and perceptive), provides the agents with a proper mechanism to communicate by names, regardless of the complexity and nature of the underlying environment (i.e. operating systems, networks, etc.).

This particular view of agents is the only assumption for analysis, while the design is specific to the JADE platform, which is essentially a FIPA-compliant realization of the above vision, i.e. in the design phase, the constructs provided by the JADE platform are assumed.

Planning

Planning is not currently in the scope of the methodology. However, some brief pointers are provided. For the planning, it is advisable to consult the literature in order to ascertain for which problem domains an agent-based solution has been applied effectively and what are the potential pitfalls associated with agent-based development. By doing so, it will save the designer a lot of time and effort, by preventing the unnecessary implementation of an agent-based solution, or, on the other hand, confirming the appropriateness of an agent-based solution, for a particular case.

Analysis

The analysis aims to clarify the problem to a sufficient level of detail, with minimal concern about the solution. There are six steps in the analysis phase which are iterated until a sufficient model of the system is obtained. The important elements gained from carrying out the above steps are the artefacts. These artefacts form the basis for the design phase.

Design

Once the analysis has been carried out, a move is made to the design phase, which aims to specify the solution. The solution focuses on the JADE platform. It is possible to move back and forth between the analysis and design whenever necessary. There are a total of ten steps in the design phase which are iterated.

Implementation

Like planning, implementation is not currently in the scope of the methodology. Note, however, that for the implementation phase, it should be very straightforward in general once the artefacts from the analysis and design stages have been realised. The system designer should proceed by consulting the rich set of tutorials available on the JADE website (JADE 2006).

MODELLING THE INTEGRATION OF OPERATIONAL TASKS USING AGENTS

This section gives details on why this particular problem lends itself to an agent-based solution, and then goes on to give details of the implementation of the technique in the pilot plant at Murdoch University.

Why was an Agent-Based Solution Adopted?

It has been pointed out and demonstrated in the past that agent-based software engineering is especially appropriate as an integration technology (Genesereth and Ketchpel 1994). A major benefit of an agent-based solution is flexibility and adaptability, not present in the techniques located in the “Previous Works” section. These benefits, in addition to the fact that the integration of operational tasks in a chemical plant naturally lends itself to an agent-based solution, will be discussed below.

Flexibility results because an agent-based solution allows heterogeneity – programs written by different people, at different times, in different languages, running on different operating systems, can be easily assimilated into a multi-agent system. This means that already existing software is permitted (via agent encapsulation) and not wasted, leading to time and cost savings. The flexibility results from the adoption of a universal agent communication language (ACL). The ACL allows heterogeneous programs to communicate with each other using a common language with agent-independent semantics. Agent development platforms such as JADE allow definition of a content language, which allows the agents to communicate using languages such as XML and RDF, in addition to the standard SL and LEAP. Likewise, an agent-based solution can be developed with the user (in this case plant operator) in mind. Flexibility is seen as a very important factor for integration of operational tasks in a chemical plant, since there is a very high possibility that the software components present in the system will use
varied solution methods, and the agent-based solution quickly and easily accounts for this.

Adaptability results because new software components can be easily added to the system in a modular fashion with little disruption to the function of the overall system and its components. All that needs to be done is to analyse and implement the dependencies between the existing agents and the newly added agent. This property of an agent-based solution is very important and desirable, since, if the solution is to be successful for the integration of operational tasks, decoupling is key, i.e. the less technical coupling there is between integrated elements, the greater the adaptability and chances for reuse. Adaptability is seen as a very important factor for integration of operational tasks in a chemical plant, since software components may be added or removed over time, and the agent-based solution quickly and easily accounts for this.

Another reason for choosing an agent-based solution is because the integration of operational tasks in a chemical plant naturally lends itself to such a solution. The mindset provided by the agent paradigm is suited to the problem for a number of reasons (in addition to flexibility and adaptability described above). Firstly, the operational tasks may be spread over several different computers. The agent-based solution inherently allows for a distributed solution. Secondly, the operational tasks are highly autonomous, but need to communicate with each other at specific times for necessary information. The agent-based solution inherently lends itself to problems with highly autonomous components that need to communicate with each other periodically. Thirdly, an operational task may be composed of several functions that must run continually (or at certain time intervals) and in parallel. The agent-based solution provides a thread for each agent, and inside the agent there are can be multiple behaviours. These behaviours run in parallel so that all the tasks are being executed continuously and automatically by the agent, without the need for human intervention. Furthermore, the agents themselves are programmed as individual threads so they can run in parallel, and there can be multiple agents on a single computer if desired. Fourthly, some of the operational tasks may require intensive data processing (e.g. for a large chemical plant). An agent (or several agents) can process these large amounts of data and communicate meta-level results to other agents whenever necessary. The above points show that the agent-based solution is suitable for this particular problem domain. As already mentioned, though, an agent-based solution may not always be the best solution tool for other problem areas, and it is important to thoroughly analyse the problem before proceeding with this technique.

**Pilot Plant Implementation**

The following modules that carry out the operational tasks were developed: data reconciliation; monitoring; fault detection; fault diagnosis; and supervisory control. The following modules were pre-existing (via Honeywell SCAN 3000): data acquisition and regulatory control. Furthermore, the scheduling and planning have not been implemented due to the complicated nature of these operational tasks. The abovementioned modules have deliberately been developed using different techniques (and programming languages), including: neural networks, sequential quadratic programming (using CFSPQ which is programmed in the C language), and standard Java methods.

The pilot plant representing the Bayer Process at Murdoch University was used as a base for testing the agent-based approach to integration of operational tasks in chemical plants. The focus of the implementation was on two (of the four) main sections of the pilot plant, namely the separation and heating (see Figure 2). More details on the experimental setup can be found in Power (2004). The pilot plant control room was setup as follows: there were a total of seven computers all connected to each other as part of a local area network; six of the computers were PCs running Windows NT, while the seventh computer was a laptop running Windows XP.

**Figure 2: Separation and Heating Sections of the Pilot Plant**

The steps in the methodology (see “Methodology Summary” section above) were applied and the multi-agent system developed. The modules developed (data reconciliation, process monitoring, fault detection, fault diagnosis, supervisory control, and data acquisition) were initially spread over the six PCs, with the laptop assigned to the operator. Each of the six PCs has an agent residing on it, which is responsible for its corresponding module, and is named after that module (e.g. the data reconciliation agent is responsible for the data reconciliation module). In every case, the transducer approach was used for interactions between the agent and its corresponding module. Furthermore, each agent has complete control over its module and can start and stop execution of the module (and its individual functions) whenever desired, and in most cases obtains important information by reading a text file produced by the module. The operator agent resides on the laptop and has the responsibility of starting/stopping/killing the other agents in the system and presenting an aggregated view of the operational tasks status to the operator via a graphical user interface (see Figure 3). Each individual agent also has a simple graphical user interface which shows the status of its corresponding module, in addition to the status of the agent – Figure 4 shows the graphical user interface for the data reconciliation agent.
The results obtained from running the system indicate that the agents can adequately account for and control their corresponding modules, in addition to exchanging messages with the other agents in the system (all without any human intervention). Thus, automation and interoperation between the modules (operational tasks) is effectively accomplished via the agent layer.

Figure 3: Operator Agent GUI – Aggregated View of Operational Tasks Status

Figure 4: Data Reconciliation Agent GUI – Overview of Data Reconciliation Operational Task

CONCLUSIONS

It has been demonstrated how modelling the integration of operational tasks in a chemical plant can benefit from the use of agent-based software engineering techniques. The agent-based technique provides the designer with a suitable mindset for analyzing and designing the system, and allows for flexibility and adaptability. The successful application of the recently proposed JADE multi-agent system development methodology for integration of operational tasks in the pilot plant at Murdoch University demonstrates that this is both a natural and promising technique for this and similar problems.

ACKNOWLEDGEMENTS

The authors would like to thank the Australian Research Council (ARC), Parker Centre, and Murdoch University for their support.

REFERENCES

DISTRIBUTED SIMULATION
TOWARDS DEVS META LANGUAGE

Vladimír Janoušek
Petr Polášek
Pavel Slavíček
Faculty of Information Technology,
Brno University of Technology
Božetěchova 2
612 66 Brno, Czech Republic
E-mail: {janousek|polasek|slavicek}@fit.vutbr.cz

KEYWORDS
DEVS formalism, simulation model, meta-model, model specification, XML, XSL transformation

ABSTRACT
The aim of this article is to present DEVSML (DEVS Meta Language) that is intended for implementation of simulation models based on the DEVS formalism. A model implemented by DEVSML is independent of the concrete simulation environment implementation. Furthermore, a model can be simply and automatically transformed and used by potentially whatever DEVS-based simulation environment, such as DEVS/C++, DEVSJava, etc. Since models are generally not portable between different simulation environments, it is necessary to implement models for every framework, which costs a lot of time and resources. DEVSML eliminates this annoyance and enables creation of models, which can be shared between different simulation environments. DEVSML defines model’s structure as well as behavior by XML (eXtensible Markup Language). Transformation of models is based on the XSL transformation. For every simulation environment, an XSL template needs to be created and used for the transformation.

INTRODUCTION
The interest in modeling and simulation is still increasing, since modeling and simulation is used in many areas of human activity and cannot be substituted during design and analysis of complex systems. Actually, there is a pragmatical trend to group models and simulators into tightly linked packages. While creating a new simulation environment, basic models are reconstructed and reimplemented, since they cannot be shared among different frameworks. It seems to be desirable to specify a standard for implementations of models, which will put away this annoyance and allow migration of models between different environments, saving time and effort spent on implementation. Then the creation of libraries consisting of commonly used components and models will be possible.

The article is focused on the DEVS formalism (Discrete EVent System specification) (Zeigler et al. 2000) representing a formal basis for specification of discrete event systems. In theory, the DEVS models are independent of the chosen simulator as well as of the experimental frame. Actually, many implementations of this formalism exist, such as DEVS/C++, DEVSJava, etc. Their main disadvantage is impossibility of sharing models between them because the models are usually implemented in the same language as simulator itself.

The aim of this article is to present our approach trying to solve this problem. We are developing a meta-language for description of DEVS models that is based on XML. Models described by this language can be simply transformed to different simulation environments and frameworks without needs of their changes. Furthermore, to simplify the implementation, we are developing a prototype of a modeling tool, based on this language, which enables us to graphically specify a model with the definition of transition and transformation functions of atomic models.

The paper is organized as follows. The first section shortly reviews basic components and terms of DEVS formalism. Next section delimitates the areas of our work with respect to the DEVS standardization group. It deals with existing tools and discusses their advantages and disadvantages. The main parts of the article are the last two sections, describing the DEVSML and shortly reviewing our modeling tool based on this meta-language. The advantages our approach gives and future plans are discussed in the conclusion.

THE DEVS FORMALISM
The DEVS formalism specifies discrete event systems hierarchically. We start from the atomic models, from which larger coupled models are built. An atomic model M is defined as:

\[ M = \langle S, \tau_a, \delta_{int}, X, \delta_{ext}, Y, \lambda \rangle \]

where
\( S \) is the sequential state set,
\( \tau_a: S \rightarrow R_{\geq 0}^\ast \) is the time advance function,
\( \delta_{int}: S \rightarrow S \) is the internal transition function,
\( X \) is the set of external input event types,
\( \delta_{ext}: Q \times X \rightarrow S \) is the external transition function.
where $Q = \{(s, e) | s \in S, 0 \leq e \leq ta(s)\}$,

$Y$ is the set of external event types generated as output,

$\lambda : S \rightarrow Y$ is the output function.

The coupled model can contain atomic models and also coupled models, connected together. This model can be inserted as a component in another coupled model, forming a hierarchical structure. A coupled model $C$ is defined as:

$$C = \{X_{self}, Y_{self}, D, EIC, EOC, IC, select\}$$

where

$X_{self}$ is the set of external input events,

$Y_{self}$ is the set of output events,

$D$ is a set of DEV5 component models,

$EIC$ is the external input coupling relation,

$EOC$ is the external output coupling relation,

$IC$ is the internal coupling relation,

$select$ is a function, the tie-breaking selector.

More detailed descriptions for the definitions of DEV5 models, their variants and abstract simulators can be found in (Zeigler et al. 2000).

EXISTING SYSTEMS AND TOOLS

There exist many implementations of DEV5 formalism. Some of the most known are DEV5C++ (Zeigler et al. 1996), DEV5Java (Sarjoughian and Zeigler 1998; Zeigler 1997) and we can also mention PythonDEV5 (Bolduc and Vangheluwe 2001) among others. DEV5Java is considered as a reference implementation among them.

There exists an effort to develop a standard of the DEV5 formalism. One team that is concerned with the standardization is the DEV5 standardization group. Their research (Wainer 2005) can be divided into three basic areas: interoperability of existing DEV5 tools (1), specification of the minimal ‘kernel’ for a tool to be DEV5 compliant (2), definition and implementation of a language for atomic and coupled model definition and broadening the models between the community of users (3).

There exist some articles (Fishwick 2002) concerning the area (2) and proposing the manners of simulation models description. The description or model’s structure is based on XML, the translation functions are dismissed or are described with a pseudo code.

There also exists a general modeling tool ATOM3 (Lara and Vangheluwe 2002) focused on meta-modeling and model-transforming. Meta-modeling refers to description or modeling different kinds of formalisms used to model systems. Model-transforming refers to the automatic process of transforming a model in a given formalism to another one (in the same or different formalism). The tool is based on graph grammars and uses graph rewriting for model transformation.

Our focus is the area (3). At least two similar projects are connected with this area. The first is DEV5W (Yung-Hsin and Lung-Hsiung 1998). In this project, the model’s structure description is based on XML and the description of behavior of atomic models is specified by a pseudo code. An elegant solution represents the second project (Schäfer 2003) that uses XML only. The description of the functions is done by rules with a finite set of states so only finite state automata can be specified by this approach.

DEV5ML

The primary motivation to develop the DEV5ML was to enable portability of a model between our experimental simulation tools and other DEV5-based simulation frameworks. The portability of a DEV5 model allows us to use advantages of complementary features of the other frameworks. However, portability is a general problem and its solution can introduce interesting possibilities and facilitate the modeler’s effort.

Structure of a Model in DEV5ML

DEV5ML (DEV5 formalism Meta Language) is based on XML (extensible Markup Language). An XML document has a hierarchical structure so that the hierarchical structure of DEV5 model can be mapped to XML easily. An XML document defining a coupled DEV5 component comprises an input and output ports specification, a list of names of the inner components (i.e. atomic DEV5 models) - with links to the documents containing their definitions - and a description of couplings of the components. Every DEV5 component is defined in a separate (possibly detached) XML document, which can be stored in local files as well as anywhere on the internet. This way it is possible to create publicly available libraries of reusable components. An example of a coupled component defined by DEV5ML is shown in Figure 1.

Figure 1 depicts also a definition of an atomic model. The definition of input and output ports of an atomic component is similar to the ones of a coupled component. A concept of a super model can be applied here to simplify the implementation by means of inheritance - the definition of ports, state variables, and functions can be inherited from a super-model.

As the definition of an atomic and a coupled DEV5 model structure is quite simple, the definition of the atomic DEV5 behavior (specified by the internal transition, the external transition, the output function and the time advance function) is a bit more problematic. We have got inspired by JavaML project (Badros 2000), which defines an XML representation of Java code. This way, the Java code is available for optimization, verification, and/or transformation. JavaML features perfectly fit to our needs here. That is why our notation for basic programming
constructs and expressions usable in DEVS functions specification came out directly from JavaML.

![Coupled DEVS Model](image)

```xml
<coupled name="coupledTest" modelX="50" modelY="50">
  <port>
    <port name="input1"/>
  </port>
  <input/>
  <output/>
  <port name="output1"/>
</output>
</coupled>
```

The hierarchical structure of a model and the coupling of the components can be specified graphically. This is shown in Figure 1. Such a visual language can be transformed to DEVSMOL quite simply. The graphical specification of models and their transformation is one of the main features of our experimental modeling tool we are developing. This is discussed in the next section.

What is more difficult is the behavior specification of the atomic DEVS. Some simple language or a pseudo code can be used for specification of the functions. Currently, we are experimenting with a Lisp-like language, which is shown in Figure 2.

**LISP-LIKE CODE:**
```lisp
(if (> A 10)
    (for i from 0 to 5 do
      (set B[i] 0))
    (set A 5))
```

**DEVSMOL:**
```xml
<Code>
  <if>
    <Binop operator=">">
      <Left>
        <GetVar name="A"/>
      </Left>
      <Right>
        <Literal value="10"/>
      </Right>
      <Binop>
        <Test>
          <Code>
            <Loop kind="for" variables="i" start="0" end="5" step="1">
              <GetStateVar name="B[i]"/>
              <Literal value="0"/>
            </Loop>
          </Code>
        </Test>
      </Binop>
    </if>
    <Else>
      <Else>
        <GetVar name="A"/>
        <Literal value="5"/>
      </Else>
    </Else>
    <GetStateVar name="A"/>
    <Literal value="5"/>
    <GetStateVar name="A"/>
    <GetStateVar name="A"/>
  </Code>
</Code>
```

Figure 2: Lisp-like Code and Corresponding Representation in DEVSMOL.

Besides a translator from such a source language to the DEVSMOL code it is essential to have a possibility to translate the DEVSMOL code to the source code as well. Then we can use DEVSMOL code as the only code, which have to be stored and maintained. We also suppose that there can be even more languages used as the user-friendly views on DEVSMOL-specified model.

To be more consistent with the graphical specification of DEVS components structure, one of the possibilities we are going to investigate comprise visual languages for expressing the DEVS functions. We have slightly experimented with a visual language based on the UML (France et al. 1997), especially the diagram of activity. An example of this approach is showed in Figure 3 (code from Figure 2 is used). Nevertheless we consider this approach to be very experimental.

---

**Source Languages for DEVSMOL**

The implementation of DEVS models directly in DEVSMOL is possible but not very handy. A more user-friendly language has to be chosen for modeling.
A MODELING TOOL

The prototype implementation of a DEVSML-based modeling tool is being implemented in the Java. The whole system has two main parts. The first part represents a graphical user interface used to create the structure of a model and a definition of its atomic models behavior. The second part of the system supports transformation of a model defined in DEVSML into an equivalent implementation for some simulation tool or environment.

The model structure editor allows for insertion of components whose definition can be stored in some local files, or somewhere on the internet. Furthermore it allows for the creation of new atomic components with the use of inherited properties of other atomic components. Among the atomic DEVS functions, the tool supports also user-defined functions that can be called from the atomic DEVS component's functions specifying its behavior. User defined functions are included as properties of atomic components.

The transformer is not just an implementation of XSL transformations. Its task is also to retrieve from internet or from files the necessary DEVSML definitions needed for the transformation and after that to link and to sort the results of the XSL transformation in a form acceptable by the chosen target simulation framework.

CONCLUSION

The article deals with sharing of simulation models among different DEVS-based simulation frameworks. We have presented a proposal of our solution - DEVSML. A model defined by DEVSML is independent of the particular implementation of the DEVS simulator. DEVSML specifies the hierarchical structure of model as well as the structure and the behavior of the atomic components by XML. The creation of models is simplified with the use of our modeling tool prototype. This experimental tool allows us to graphically specify the model structure and its behavior. As a future work, we are planning to extend it by support for more simulators and to implement new data types useful for implementation of atomic models behavior.

DEVSML puts away the dependence between the implementation of a model and a simulation environment. A model specified by DEVSML is portable to theoretically whatever DEVS-based simulation framework which is the main gain of this project. The benefits of portable models are very important. For example, the development of a model can be done in a sequential framework where testing is often easier. Afterwards, the model can be ported and simulated in a distributed or real-time simulation framework that could be more appropriate for some particular simulation studies. DEVSML can be used also for creating libraries of commonly used reusable models.

The next benefit of DEVSML is the possibility of automatical verification of new simulation environments, as suggested in (Wainer 2005). A verification can be done through a formal proof, which is obviously difficult. The
other method of verification is to simulate model in a reference simulation environment and than compare the results with simulation in the tested environment. DEVSM offers sharing of models between the tested and reference environment.

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


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.

PETR POLÁŠEK received the M.S. degree from 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 meta-modeling.

PAVEL SLAVÍČEK received the M.S. degree from Faculty of Information Technology, Brno University of Technology in 2003. He is a last year Ph.D. student in the Department of Intelligent Systems at the Faculty of Information Technology, Brno University of Technology. His research focuses on meta-modeling, distributed simulations and multiagent systems.
AN EVENTS SYNCHRONIZATION APPROACH FOR INTEGRATION OF SIMULATORS IN A DISTRIBUTED ENVIRONMENT

Moreno Carullo  
Antonella Zanzi  
Ignazio Gallo  
Alberto Coen-Porisini

Dipartimento di Informatica e Comunicazione  
Università degli Studi dell’Insubria  
Via Mazzini 5, 21100 Varese, Italy  
E-mail: \{antonella.zanzi\|ignazio.gallo\|alberto.coenporisini\}@uninsubria.it

KEYWORDS
Events synchronization, distributed discrete event simulation, simulators integration, open-source software.

ABSTRACT
In this paper the issue of events synchronization in the Simulator Integration Platform (SINPL) is presented. SINPL is an open-source software platform aimed at supporting the simulation design and the integration of heterogeneous simulators in a distributed environment. The paper introduces the platform and focuses on the approach used to manage the events synchronization.

INTRODUCTION
In the field of simulation the relevance of distributed architectures is increasing in order to reuse existing simulators and to model complex systems. Existing solutions 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. The Discrete Event Simulation (DES) can be classified according to the way in which simulation time advances. In time-driven discrete simulation, simulation time is incremented by a constant time step, while in the event-driven approach, the increment of simulation time is triggered by the next earliest occurring event. It is well known that, for general-purpose simulation, the event-driven approach is better suited than the time-driven one (Ferscha 1995).

In this work, event driven DES is performed in a distributed manner; in particular, this work focuses on the issue of events synchronization in the SINPL platform. Synchronization approaches broadly fall into two categories (Ferscha 1995; Fujimoto 2003): conservative and optimistic, according to the ways they adhere to the local causality constraint. Conservative approaches (Chandy and Misra 1979) avoid all possible causality errors by strictly adhering to the local causality constraint. Optimistic approaches (Jefferson 1985), on the other hand, guarantee to detect any occurrence of causality errors, but require a rollback mechanism to restore the simulation to a correct state.

The conservative approach has been chosen since the SINPL platform aims at integrating existing simulators, which may not be provided with rollback capabilities.

The paper is organized in the following way: first the SINPL platform is introduced, then the issue of events synchronization is presented followed by the description of the solution adopted. Finally, some conclusions are drawn.

THE SINPL PLATFORM
The SINPL platform provides the necessary capabilities for 1) integrating heterogeneous simulators in a distributed environment, 2) supporting both the design and the simulation activities.

The simulation design process supported by SINPL comprises three main activities described in the following.

1. Information Model definition, which means to define the elementary elements that belong to an application domain, which represent either the logical components of a system or the simulators. Thus, each domain has its own Information Model. As a result, the SINPL platform can be used in many different application domains by defining the appropriate Information Models.

2. Simulation Architecture design, which means to build a system by instantiating the elements of the Information Model. Each instance comprises input and output Gates, which are in turn linked by means of SimulationLinks. In particular, an input gate is a gate through which a component receives data, while an output gate is a gate through which a component sends data.

§ Work supported by the Italian MIUR-FIRB – Tecnologie abilitanti per la Società della conoscenza ICT.
The Simulation Architecture provides a logical view of how the different simulation models cooperate by defining both the data flow description, that is, which data are exchanged, and the control flow description, that is, the way in which the simulators interact.

The semantics of the Simulation Architecture is given in term of High Level Petri Nets (HLPN) (Jensen et al. 1997) in which data are associated with tokens and actions with transitions. Each component defined in the Simulation Architecture is associated with a HLPN, and thus the Simulation Architecture itself results in a HLPN obtained by composing the different HLPN associated with the components therein.

3. Simulation Architecture execution, which means defining how the simulation has to be carried out. This requires first to define the input data needed by the different simulators and then to actually execute the simulation. The HLPN associated with each component describes the available operations and their relationship with input and output gates. Therefore, executing the Simulation Architecture corresponds to executing the HLPN. When a transition (and consequently the associated action) is enabled, the input data is taken from the input places and sent to the real simulator; similarly the output data produced by the simulator, is used to update the corresponding HLPN.

SINPL tools

The platform provides a set of tools to manage the simulation lifecycle. Figure 1 shows the deployment diagram of the SINPL platform.

Figure 1: SINPL – Deployment diagram

The definition of the Information Model can be done using the Infocreator tool, while the graphical editor System Editor (SED) is the tool aimed at the definition of the Simulation Architecture.

The Distributed Simulation Controller (DSC) is the tool in charge of the execution of the simulation itself, managing the communication among the different simulators. The DSC coordinates the simulation execution among the distributed software components: it is in charge of determining the control flow by executing the HLPN associated with the Simulation Architecture and managing data exchange among simulators.

The communication between the DSC and the simulators is implemented using CORBA. Moreover, integrating different simulators requires common interfaces and data structures. As a consequence simulators may need to be accessed through a software adapter.

All the tools have been developed using Java and are open-source.

EVENT SYNCHRONIZATION

Different ways of decomposing a simulation for processing on multiple processors are present in literature (Righter and Walrand 1989; Ferscha 1995; Ferscha 1996). In this paper we follow the Distributed Events or Logical Process Simulation (LPS) approach with a decentralized event list (EVL), whose basic architecture is reported in Figure 2.

Using this approach, the representation of the system under simulation can be divided in n logical processes LP1, ..., LPn, that interact among them by exchanging event messages through a communication system. Each logical process simulates a portion of the whole system and is composed of a simulation engine and a communication interface. The simulation engine controls the execution coordinating the events flow by means of event queues and timers. Two kinds of events can be processed: internal and external. Internal events represent state changes in a LP, while external events represent time-stamped messages sent to/received by other LPs. Time-stamped messages are of the form \textit{evt}@t, where \textit{evt} denotes the event and \textit{t} is the associated timestamp, representing the value of the \textit{Local Virtual Time} (LVT) of the sending LP when the message was created and sent. The communication interface of each LP is composed of one input buffer and a channel clock for every channel connected to another LP. In the input buffers the arriving messages are stored in FIFO order.

This model allows the simulation of a system in a distributed way and since the LVT of the different LPs may be different the causality constraint (Fujimoto 2003) must be guaranteed in order to ensure correctness.

For example, let \textit{evt}A and \textit{evt}B denote two events with timestamps \textit{t}A and \textit{t}B respectively such that \textit{t}A < \textit{t}B. Assume \textit{evt}A is coming from simulator \textit{S}A, \textit{evt}B from simulator \textit{S}B, and let simulator \textit{S}A be the target of both events. Thus, in order to ensure the causality constraint, \textit{S}A should process first \textit{evt}A and then \textit{evt}B. However, since the “actual order” in which \textit{S}A receives the events may be different from “simulation order”
of events, a correct synchronisation is necessary to avoid any violation of the causality constraint. The approach used in this work to guarantee the causality constraint is based on the Chandy-Misra-Bryant (CMB) protocol (Chandy and Misra 1979). In the CMB protocol an event evr can be processed by a LP only when it can be considered safe, that is when evr has a timestamp such that the LP is guaranteed not to receive external event messages with smaller timestamp. This is realized by introducing the concept of Local Virtual Time Horizon (LVTH) in each LP. The LVTH represents the time value until LVT is allowed to progress since no external events can arrive with a timestamp smaller than LVTH. In this way the CMB protocol implements a blocking until safe to process policy.

Notice that the use of such a policy can, in some cases, generate a deadlock. Several approaches have been proposed to overcome this problem such as the definition of lookahead (Chandy and Misra 1979) and the introduction of empty messages, called nullmessages (Misra 1986). The lookahead represents the minimum time interval between an event and the next one for a given communication channel. Therefore, the lookahead depends on the underlying simulation model. A nullmessage is a special message of the form @i that is used only to prevent deadlock. When a nullmessage is sent, this means that no other messages with smaller timestamp can be sent in the future. Therefore, a nullmessage is not related to the simulation model and it is used only for synchronization purposes.

**TIME SYNCHRONIZATION IN SINPL**

As described before, DSC makes use of HLPNs to model data exchange between simulators. In order to manage time, each transition is associated with a minimum and a maximum firing time along with any possible firing condition. Moreover, each token is associated with a timestamp.

In SINPL each simulator belonging to a given Simulation Architecture is viewed as a logical process. Thus the SimulationGateLinks connecting the SimulationGates of the different simulators are viewed as the communication channels that connect the different LPSs.

Figure 3 shows an example of the correspondence between a SINPL Simulation Architecture and the LPS model. Gates G0 – G7 are SimulationGates components connected among them by means of SimulationGateLinks components. Gates G1 and G3 are connected through a communication channel, as well as gates G2 and G7, and G4 and G0.

According to the LPS model in which each LP represents a portion of the system, each simulator is associated with a HLPN. Moreover, each SimulationGate belonging to a given simulator corresponds to a place of the associated HLPN.

A SimulationGateLink (i.e., a communication channel) connecting two SimulationGates is modelled by means of a transition connecting the places corresponding to the SimulationGates. Such transitions are treated as instant transitions.

SINPL events are of two different kinds: internal and external. Internal events represent the firing of scheduled transition and the production of tokens used to update the HLPN marking, while external events represent the arrival of tokens coming from other simulators and nullmessages.

![Figure 3: SINPL architecture and LPS correspondence](image)

Figure 4 shows the HLPN (only partially depicted) and the input buffers corresponding to the example of figure 3. Since the communication channels are treated as instant transitions; a token in place o1 (o2) will be instantly queued for insertion in place i1 (i2). For instance, if the token in o2 has a timestamp equal to 4 then an external event for i2 is generated with timestamp equal to 4. Moreover, a nullmessage with timestamp 4 is generated and sent to i1. Such nullmessage is required to notify Sim3 that the simulator Sim1 has produced an event with timestamp equal to 4 (represented by the token sent to Sim2). In this way Sim3 knows that any further event coming from Sim1 will have a timestamp greater than 4.

![Figure 4: External events](image)

Moreover, it is possible to provide more information about when the next event will be produced by using the lookahead. The lookahead is associated to each communication channel and thus in our approach it is associated with each output place of the HLPN. Its value is determined statically at start-up using HLPN transition times and then it is updated, at
runtime. Figure 5 shows more in detail simulator Sim1 by providing information on the transitions that can produce tokens in places \( o_1 \) and \( o_2 \). Thus, the lookahead of output channels \( o_1 \) and \( o_2 \) depends on the timings of transitions \( T_1 \), \( T_2 \), and \( T_3 \).

Let us consider the output place \( o_1 \) whose input transitions are \( T_1 \) and \( T_2 \). The minimum value between the minimum firing times is 2 and thus the initial value of the lookahead of \( o_1 \) equals 2. In the same way one can determine that the initial value of the lookahead of \( o_2 \) is equal to 3. Therefore, in the previous example, if we take into account the lookahead of the channel associated with the output place \( o_1 \) then the timestamp of the nullmessage becomes 6, that is 4 + 2 (lookahead).

![Diagram](image)

**Figure 5: Full timed transitions and output channels**

Formally, the initial value of the lookahead associated with an output channel is given by the following formula (1):

\[
\text{lookahead}(\text{place}) = \min\{ ts_{\text{min}}(t) \mid \text{place}, \in \text{post}(t) \}
\]  

(1)

where \text{post}(t) is the set of the output places, and \( ts_{\text{min}} \) is the minimum firing time. If no information about firing time is available for at least one transition connected to a particular place, the associated lookahead is equal to zero.

Once a token has been produced in an output place it is necessary to recompute the lookahead. For instance, let us assume that at time \( t \) both \( T_1 \) and \( T_2 \) are enabled. Moreover assume that at time \( t+2 \) \( T_1 \) fires. Then the next event for \( o_1 \) will not occur before \( t+3 \), that is the lookahead value is now equal to 1. Instead, suppose that only \( T_2 \) is enabled at time \( t \), while \( T_1 \) becomes enabled at time \( t+1 \). Then, the first event is produced at time \( t+2 \) (the firing of \( T_1 \)), while the next event (the firing of \( T_2 \)) cannot be produced before \( t+4 \), that is the lookahead value is still equal to 2.

In this way, the lookahead is recomputed at run-time using the event list allowing the DSC to predict when transitions can fire, that is which will be the smallest value of the timestamp of the next event.

**Comparison with the CMB protocol**

The CMB protocol could be used to execute in a distributed way a Timed Petri Net. In such a case when a transition \( T \in [t_{\text{min}}, t_{\text{max}}] \) is structurally enabled (i.e., all input places contain at least one token) it is fired at least after \( t_{\text{min}} \) time units and not later than \( t_{\text{max}} \) time units. Once \( T \) is fired, token are instantly removed from all input places and put in all output places.

In SINPL each simulator is characterized by a HLPN, which models the way in which the simulator works. Thus, a transition structurally enabled represents the fact that all input data needed by the simulator are present. Therefore, the transition is instantaneously fired that is the simulator is notified that it can process the events represented by the tokens. The timings associated with the transition model how much time units are requested by the simulator to produce the output events. Thus, transitions need to be scheduled for firing whenever the required tokens are present, and then DSC has to wait until the response of the simulator arrives. As explained later on, output tokens representing the simulator response do not update immediately the HLPN marking, but are inserted in the event list.

Moreover, the way in which DSC controls data exchange among simulators is derived from the CMB protocol. However, there are several differences between our approach and the CMB protocol, which are discussed in what follows.

The CMB protocol does not allow concurrent execution within a single logical process. Since simulators can execute multiple operations concurrently, it is necessary to modify the protocol in order to allow concurrent transitions to occur. Given a transition \( T \), with timings \( t_{\text{min}} \) and \( t_{\text{max}} \), let us assume that \( T \) fires at time \( t_1 \) (i.e., the LVT of the LP executing \( T \) equals \( t_1 \)). Therefore, \( T \) will end at time \( t_2 \) such that \( t_2 \geq t_1 + t_{\text{max}} \) and \( t_2 \leq t_1 + t_{\text{min}} \). As a consequence, it is necessary to prevent the processing of any event having timestamp greater than \( t_2 \), in order to guarantee the causality constraint. In other words, handling concurrent transitions (that is, simulator operations) requires one to prevent out-of-order event processing. In SINPL we decided to face this problem by introducing a new kind of (internal) event, which we refer to as a semaphore event, whose aim is to limit the processing capabilities of LPs.

When a transition is fired a new semaphore event is inserted in the event list with timestamp equal to LVT plus the minimum transition firing time (\( t_{\text{min}} \)). When the semaphore event is processed, the logical process will wait until the transition has ended, that is the simulator replied, preventing in this way future events to be processed before simulator reply. Notice that between transition firing and semaphore event processing the logical process can select and execute other events, including other transitions firing.

Figure 6 shows an example in which the initial LVT is equal to 7 and the next scheduled event ev is at timestamp 9. Thus, when LVT equals 9 ev is processed, a token is put in place \( i_1 \) and transition \( T_2 \) can fire. When \( T_2 \) fires, the simulator is invoked in order to perform the actions associated with \( T_2 \) and a semaphore event is inserted in the event list with timestamp 11 (i.e., the value of LVT plus the minimum transition time).

The next event is the token arrival in \( i_2 \), with timestamp equals to 10, which enables transition \( T_1 \). Since this event has a timestamp less than the timestamp of the semaphore event, it can be processed, that is \( T_1 \) fires. Notice that if its timestamp were equal to 12 then the semaphore event would have prevented \( T_1 \) from firing. When \( T_1 \) ends the semaphore is removed and the next events can be processed.

Notice that semaphore events are also used to prevent the generation of nullmessages that could violate the causality constraint. In fact according to the CMB protocol, processing the token in \( i_1 \) would cause a nullmessage with timestamp 13 to be sent through \( o_2 \). However, the semaphore event inserted by \( T_1 \) prevents the generation of such nullmessage, which could cause a violation of the causality constraint.
A second difference concerns the way in which nullmessages are generated by simulators having either no input channel or input channels with no simulator connected. Notice that such simulators can only generate events without interacting with other simulators. Thus, when they stop producing events they send a nullmessage whose timestamp has value “infinite” and then they can quit. This nullmessage notifies the connected simulators that they will not receive anymore event in the future.

A third improvement was done to prevent unnecessary waiting and excessive nullmessage generation. Whenever processing an event DSC does not take into account all input channels in which there is an event with timestamp equal to LVT. This can be done since there is enough information to predict that the next events on those channels will have a timestamp greater than the one currently processed.

CONCLUSIONS

SINPL is a software platform aimed at supporting simulation design and integration of simulators in a distributed environment. In this paper we discussed the way in which events synchronization is carried out in SINPL.

The approach has been tested with an example in the Flexible Manufacturing System context and has proved to be effective. A Flexible Manufacturing System (FMS) is composed of several machines connected by means of a transport system. The transport system carries the raw parts to the machines where they are processed. Once the machines have finished their job the parts are moved back to the load station where they are unloaded. Moreover, the machines use a tool-room as a repository for the tools they need in order to properly work the raw parts. A computer controls the machines and the transport system (Matta et al. 2004, Upton 1992).

Using a distributed simulation in the FMS field provides some advantages. It is possible, for example, to solve the problem of confidentiality in the context of a supply-chain with external supplier. Moreover, a distributed simulation provides the possibility of simulating multiple levels of manufacturing systems at different degrees of resolution, creating an array of low-cost simulation models that can be integrated into larger models.

In the simplified FMS example we used to test the platform there are two flexible machines that work on pallets coming from a buffer station and a tool room simulator that works as instruments provider. Machines work on raw pallets with appropriate tools and the resulting finished parts are put back on the buffer station.

Each simulator sends and receives messages that represent requests for loading/unloading a pallet or for getting/putting back a specific tool from/to the tool room.

The buffer and the machine simulators are connected by means of two communication channels. Ad hoc simulators have been written in Java and have been extended to support the required IDL interfaces.

The example, although simple, is meaningful since pallet and tool requests rely on correct synchronization among simulators.

Future work regards the communication between simulators and the DSC module. Work has been started to allow coexistence of the CORBA communication system and the webservice SOAP protocol. In this way a distributed "web" simulation is feasible, deploying entities on different remote sites, without the technical difficulties that could arise with CORBA ORBs (in particular, firewalls and listening ports).

REFERENCES


AI-BASED SIMULATION APPLICATIONS
Rewarding and Punishing Evolving Collusion in a Simulation of an Oligopolistic Marketplace

Stephen Huxford and David Hunt
Faulty of Information Technology
Monash University
Wellington Road CLAYTON 3800, Australia
E-mail: {Stephen.Huxford|David.Hunt}@infotech.monash.edu.au

KEYWORDS

Genetic Algorithm, Simulation, Marketplace

ABSTRACT

The time phased behaviour of participants with interdependent pricing strategies in a marketplace is simulated using a GA. Each member of the population embodies a pricing strategy and their fitness equates to their profitability. Population members are monitored for collusive behaviour which is illegal in many real marketplaces because it is perceived to unfairly boost the profits of the colluders. In many of these same marketplaces a neutral authority has the power to punish collusion when detected. The marketplace simulation incorporates both the benefits and rewards of collusion. An investigation of the effects of varying the relative magnitudes of benefits and rewards and the time lag between the reaping of rewards and imposition of penalties has been made.

INTRODUCTION

An oligopolistic marketplace is one in which relatively few sellers compete to maximize their profits by selling different brands of the same product. In addition the marketplace is such that the actions of any given seller will influence the actions of the remaining sellers. This interdependence makes it a non-trivial exercise to discover selling strategies that produce market-leading profits over time.

One promising approach to uncovering such strategies is to evolve a simulation of the marketplace. If a representation of a time-phased selling strategy can be codified then several of these strategies can be allowed to compete for the profits available from the marketplace. If we begin with random strategies the most profitable strategies can be selected after a marketplace simulation. These strategies can be subject to genetic operations and allowed to compete in a second marketplace simulation. Repeating this cycle of simulation, selection of the fittest (most profitable) and genetic operation should encourage the evolution of profitable strategies.

Although retail marketplaces often involve complexities such as buyer segmentation and the effects of advertising, some important markets are far simpler. For instance, in the Australian petroleum retail marketplace profits depend almost exclusively on a seller’s current price point and this price point’s related market share. In addition the marketplace is oligopolistic. To protect buyers, Government regulations forbid price collusion amongst petroleum sellers. Despite or because of these regulations more than 30 inquiries into the pricing strategies of the petroleum sellers have been held over the last twenty five years. Any insights that an evolving simulation of such a marketplace can provide, especially with respect to pricing collusion, would be valuable.

SIMULATION AND GENETIC ALGORITHM DETAILS

A marketplace simulation (henceforth known as a game) proceeds as a series of time steps. At each time step sellers (henceforth known as players) take as input their price rank at the last two time steps. Their output is a price movement which may or may not create a new price rank for them at the current step. A player’s price rank at the current time step will subsequently influence their price movements for the next two time steps since it will be part of that player’s price rank history. Players begin a game with a common initial price. For four players there are 20 possible price rankings and therefore $20^2$ possible two step price rank histories. More generally, if there are n players then the number of price rank histories is:

$$\sum_{k=1}^{n} \frac{n-1}{k}$$

The example presented in Table 1 clarifies the concept of a price rank history where each price rank includes the relative position of a player’s price and the ranking structure of all players’ prices.

<table>
<thead>
<tr>
<th>Time Step</th>
<th>Prices (player 1, 2, 3 and 4)</th>
<th>Price Rank, player 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>t - 2</td>
<td>(59, 61, 61, 62)</td>
<td>2(1223)</td>
</tr>
<tr>
<td>t - 1</td>
<td>(60, 61, 62, 63)</td>
<td>3(1234)</td>
</tr>
</tbody>
</table>

Table 1. An Example of a Price Rank History
In each time step of a game only three price movement responses are allowed namely: increment or decrement the current price by one unit or make no change. Conversion of a player’s remembered two step price rank history to a price movement response is performed by looking up the appropriate position on a chromosome consisting of a 3-nary byte string. This string is the representation of a player’s strategy. The string is 400 bytes long and each byte can be considered as a gene that can be in one of three possible alleles: -1, 0 or +1, to indicate a price movement response. The position of a byte in the byte string indicates which of the 400 possible price rank histories it contains the response to. In practice several additional start up bytes are required to take care of responses in the first two time steps of each game when players have no remembered two step price rank history. Each gene represents an independent response component of a strategy so it can be mutated in isolation and crossover operations do not require a repair mechanism.

This representation of a strategy by a byte string suitable for processing by a GA, is an adaptation of one used by Axelrod (Axelrod 1987) to investigate the Iterated Prisoners’ Dilemma (IPD) game and an extension of this representation by Yao and Darwen (Yao and Darwen 1994) to investigate the N-person IPD.

This representation has been used previously to explore aspects of an oligopolistic marketplace other than collusive behaviour (Cheung et al. 1997a; Bedingfield et al. 1997a; Cheung et al. 1997b, Bedingfield et al. 1997b). A related representation has been described by Cheung and Hinde (Cheung and Hinde 2000).

The genetic algorithm, used in this study, typically employs a population of 20 player strategies with moderate one-point crossover and mutation rates of 0.2 and 0.01 respectively. Fitness scaling is used to both dampen extreme fitness values and emphasize smaller differences. Selection is performed using the roulette wheel approach. The algorithm is based on one presented by Goldberg (Goldberg 1989).

During each generation the marketplace is simulated \( \binom{20}{4} \) times as a game is played for each combination of 4 player strategies taken from the available population of 20. Typically a game proceeds for 22 time steps. A strategy’s fitness (profit) is accumulated over every step of every game it participates in during a generation.

**MARKET SHARE MODEL**

Profit calculations are based on price and market share which, of course, are interrelated in most retail markets. One of the more popular price/market share models amongst marketing analysts is the Multinomial Logit (MNL) model (Cooper et al. 1988; Bell et al. 1975). According to this model the market share of a player with price \( p_i \) is:

\[
S_i = \frac{e^{-\alpha p_i}}{\sum_{j=1}^{n} e^{-\alpha p_j}} \quad \text{where } \alpha > 0 \text{ is an adjustable constant.}
\]

To help visualize the effect of this model 3D profit plots for \( \alpha = 0.02 \) and \( \alpha = 0.2 \) are shown in Figures 1 and 2 respectively. To make the plots visually practical one player’s price is varied against a common price for the remaining three. Low values of alpha lead to simple market share and profit surfaces where high prices are punished only weakly with reduced market share and resultant profits. High values of alpha produce more complex surfaces that punish high prices more severely.

![Figure 1: Profit Plot for MNL Market Share Model with \( \alpha = 0.02 \)](image1.png)

![Figure 2: Profit Plot for MNL Market Share Model with \( \alpha = 0.2 \)](image2.png)

For the purposes of the current investigation a value of \( \alpha = 0.1 \) which is intermediate between these two extremes has been chosen. Our experiments have shown that such intermediate alpha values present equivocal pricing choices to marketplace participants which engender the evolution of non-simplistic pricing strategies to increase profits. Such strategies are a hallmark of real oligopolistic marketplaces.
COLLUSION DETECTION, BONUSES AND PENALTIES

As already discussed profits are accumulated by marketplace participants (players) in 4-way marketplace simulations called games. A game proceeds in time steps. At each step players decide on their price for the next step according to their pricing strategy. This strategy is based on their memory of the relative price ranking of all players in the previous two time steps.

The occurrence of collusion is tested for at each time step of each game. In this context collusion is defined as two or more players with the same price at the same time step. No distinction is made concerning the number of players colluding at a time step. However, the number of steps players contiguously collude for is monitored.

Since the strategy of a player is based only on its memory of the last two time steps it is pointless to attempt to influence the evolution of strategies based on circumstances beyond this memory horizon. Therefore the detection of collusion and sustained collusion is limited to the current time step and the previous two remembered time steps. The trigger for detection can therefore be set to be collusion on just the current time step (t1), on the current and most recent remembered time step (t2) or the current and both the remembered time steps (t3).

When collusion is detected the profits of a player for the game step are proportionately increased by a bonus (bf) and proportionately decreased by a penalty (pf). Thus equal bonus and penalty factors will cancel each other out on the same step. The increase simulates the presumed gains from collusion and the penalty simulates the actions of some neutral authority charged with monitoring and countering collusion.

Bonus and penalty factors may be symmetric (e.g. 1.25 and 0.8) or asymmetric (1.1 and 0.8) and in either case of varying magnitudes including the symmetric control case of 1.0 for both factors and asymmetric cases where penalties dominate bonuses and vice versa.

The detection of collusion for the purposes of a bonus or a penalty need not be symmetric. For instance, a bonus could be triggered by t1 (collusion after one “contiguous” step) and a penalty triggered by t3 (collusion after 3 contiguous steps). This would simulate an immediate reward for collusion but a tardy penalty.

Not all trigger combinations are worth examining. When bonus and penalty factors are symmetric and they are triggered identically they cancel each other out and are equivalent to the control case (bf = pf = 1). In addition there is no straightforward real-world interpretation of triggering bonuses after penalties.

The marketplace has been allowed to evolve for many combinations of bf, pf, t1 and t2 that are not immediately unrealistic or redundant. A sample of the insights gained are presented in the following analysis of specific marketplace simulation evolutions.

ANALYSIS OF SOME SPECIFIC EXAMPLES

In Figure 3 the best and average fitness curves over 100 generations for the control case (bf = pf = 1) are presented. In this case collusive behaviour is neither rewarded nor penalized and therefore trigger variations are redundant. Both the average and the fittest individual eventually outperform the fitness that would be achieved if no price movements were made (fitness of 100). These curves should act as a baseline reference in the following discussion of cases where collusion is punished and rewarded by various trigger variations.

![Figure 3: Best and Average Fitness over 100 generations (averaged over 10 evolutionary instances) using the MNL Market Share Model with α = 0.1](image)

In Figure 4 best and average fitness contours for four variations of bonus and penalty factors for the trigger combination t1t1 for bonus and penalty respectively are presented. This trigger combination means there is a delay between the onset of the benefits of collusion and its subsequent punishment. Figures 4(a) and 4(b) show very different contours for symmetric bonus and penalty factors depending on the magnitude of the symmetric factors. This shows that simply escalating penalties to match escalating rewards does not necessarily have a predictable effect. The remaining two plots in Figure 4 are more straightforward. They confirm that when penalties outweigh rewards (Figure 4(c)), even when the penalty’s onset is delayed, game participants cannot achieve the profits obtained in the baseline case (Figure 3). The rewards that game participants reap when bonuses exceed penalties (Figure 4(d)) can be diminished but not extinguished by increasing penalties to match rewards (Figure 4(a)).

83
In Figure 5 best and average fitness contours for three trigger variations for the same symmetric bonus and penalty factors namely $b_t = 1.43$ and $p_t = 0.7$. These contours indicate that the delay between onset of bonus and penalty is more significant than their shared delay. Thus Figures 5(a) and 5(c) show similar contours which both differ significantly from the contours in Figure 5(b). It is also interesting to note that in the cases where punishment immediately follows reward game participant’s eventually evolve to reap profits greater than the those achieved in the baseline case. However, in the case where the penalty is delayed performance is inferior to the baseline case.

Usage grids (Figure 6) are used to discover the details of the strategies of game participants as dictated by their chromosomes. These grids attempt to reveal the salient features of a chromosome. The price movement reaction of a chromosome (up, down, stay the same) to the last two remembered game participant rankings is displayed. The strategy genes are laid out in a twenty-by-twenty grid. Each grid point represents one of the four hundred possible combinations of price ranking history. Reaction is only shown if the gene has been used twice as much as expected from random use. If the use of the gene is persistent and consistent the reaction is displayed as a lowercase letter or symbol ($u$, $s$, $d$) otherwise if its use is novel it is displayed as an upper case letter or symbol ($U$, $=\, D$).

Without the imposition of bonuses and penalties associated with collusion the strategies of game players can become persistently simplistic and homogeneous after just a few generations for many settings of genetic operator and marketplace parameters. Figure 6(a) shows a usage grid for the most fit member of the fifteenth generation of an evolution. The strategies associated with such chromosomes are inherently uninteresting because of their lack of correspondence with the real world. The introduction of complexity associated with time delayed bonuses and penalties eliminates the appearance of such simple strategies during an evolution. This is clearly demonstrated by Figure

Figure 5: Best and Average Fitness over 100 generations (averaged over 10 evolutionary instances) using the MNL Market Share Model with $\alpha = 0.1$
6(b) where a far more sophisticated strategy is proving to be the fittest even after seventy generations have passed. The genetic operator and marketplace parameters are the same for both grids the only difference is the introduction of bonuses and penalties associated with collusion.

In parallel with the above work predatory strategies are being evolved by pitting members of a genetic population against a static but large number of arbitrary and semi-smart strategies rather than against each other.

REFERENCES


Goldberg, D. 1989. GAs in Search, Optimisation and Machine Learning, Reading Addison-Wesley.

Yao, X and P. J. Darwen. “An Experimental Study of N Person Iterated Prisoner’s Dilemma Games”. In Informatica 1994, Vol 18. 435-450

ONGOING AND FUTURE WORK

More sophisticated bonus and penalty factors are being developed and applied to the simulated marketplace.

Additional monitoring techniques that help understand how complex strategies, such as the ones mentioned above, achieve their success are being developed. There is a delicate balance between being overwhelmed by fine detail and burying that same detail in averages.
A MULTIMODAL CHAT-BOT BASED INFORMATION TECHNOLOGY SYSTEM

A. Augello, A. Santangelo, A. Gentile, S. Gaglio
Dipartimento di Ingegneria Informatica
University of Palermo
Viale delle Scienze
90128 Palermo
Italy
Tel: +39 0917028528
{augello.santangelo}@csai.unipa.it
{gentile.gaglio}@unipa.it

G. Pilato
Istituto di Calcolo e Reti ad alte prestazioni
CNR
Viale delle Scienze
90128 Palermo
Italy
Tel: +39 091238269
g.pilato@iacr.cnr.it

KEYWORDS
Man-machine interfaces, Speech synthesis, Speech recognition.

ABSTRACT
The proposed system integrates chat-bot and speech recognition technologies in order to build a versatile, user-friendly, virtual assistant guide with information retrieval capabilities. The system is adaptable to the user needs of mobility being also usable on different devices (i.e. PDAs, Smartphone). The system has been implemented on a Qtek 9090 with Windows Mobile 2003 and a simulation for the cultural heritage domain is here presented.

INTRODUCTION
In this work an information retrieval system in the cultural heritage domain is proposed. The aim is to offer a natural interface, easily usable also by inexpert and/or disabled users. Another target is ensure the availability of the system in environments where the mobility of the user is an essential requisite to be satisfied.

The recent improvements in information technology, the internet access availability through small, mobile devices, the advent of third-generation wireless and Bluetooth communication have totally modified the way people interact with computers. The possibilities offered by these technologies can be exploited in order to satisfy the needs of mobility. However the use of such pervasive devices presents many constraints rising from the limited dimensions of their screen; the interaction is uncomfortable because the user needs to use special styl/touch pens.

Besides the access to an information retrieval service is particularly problematic, because the user is constrained to write his requests through these limited technologies. This involves a user dissatisfaction and a large waste of time during the information research. Many systems overcome these problems offering query formulation assistance modules, including word completion as well as phrase completion and expansion (Aridor et al, 2002).

A different approach is offered by the use of a vocal interface; human speech language enables a fulfilling interaction between the user and the device.

There are many examples of systems that use spoken language interaction in Information Retrieval field to provide an user-friendly access to stored information (Lamel et al, 1995) (Gilbert and Zhong, 2003). Results deriving from the examination of the differences between using speech for input compared to traditional pen input on a PDA demonstrate that speech is more direct and allows to reduce the manual input needed during the infarction (Lyons et al, 2005).

In this paper we propose a multimodal, chat-bot based information technology system. It is accessible through a vocal interaction with a virtual assistant guide and it is usable on different devices (i.e. PDAs, Smartphone). Therefore the system easily satisfies the user needs of mobility. As a case of study, we present here an implementation of the system developed in order to assist an user during a visit to cultural heritage institutions like archaeological sites, museums, etc. In fact the use of a PDA with ad-hoc built-in information retrieval functionalities is particularly suitable for a cultural heritage environment and can help people in visiting in a natural manner the sites instead of using traditional audio/visual pre-recorded guides.

To achieve a more natural communication the use of a natural language dialogue, implemented by the means of a chat-bot is proposed. Chat-bots systems are based on a simple technology, in particular Alice (Artificial Linguistic Computer Entity) project uses a natural language processing (NLP) module that provides a natural language sentences interpretation mechanism that follows a pattern-matching scheme. The dialogue is managed exploiting the chat-bot knowledge base described as a set of question-answer modules. The structure of the knowledge base of the Alicebot chat-bot has allowed to easily induce a grammar for a speech recognition module. The particular structure of the knowledge base of the Alicebot chat-bot is well suitable to turn itself into a grammar that can be used in a speech recognition modules.

The implemented framework integrates chat-bot and speech recognition technologies in order to process and understand the user speech and give the appropriate answer.

The multimodal interface has been developed with XHTML+Voice language (Axelsson et al, 2004). The developing environment is the Multimodal Tools 4.1.2.2 for WebSphere Studio V5.1.2.

The Multimodal Tools includes the IBM WebSphere Multimodal Toolkit and IBM Multimodal Browser, respectively the developed environment and the technology for the application execution.

We have chosen this technology in order to simulate and execute the multimodal application on PDAs avoiding the need of a vocal server. The application has been implemented on a Qtek 9090 with Windows Mobile 2003. An Alice chat-bot analyzes the text recognized from the speech recognition module and gives the appropriate answer to the user. The knowledge base of the chat-bot is detailed regarding to the specific domain.

The system exploits also the properties of a well-known search engine to retrieve further information pertinent to the user query other than the chat-bot answer.
SALT AND VXML COMPARISON

The aim of this work is to build multimodal application for
an handle device in the Information Retrieval field. There are
two available technological solutions to manage this task.
The first solution uses the system architecture shown in
Figure 1 and the application is written in XHTML+SALT.
XHTML is the traditional visual web language HTML
enhanced to XML while SALT (Speech Application
Language Tags) (Wang, 2002) is a vocal language used inside
XHTML like a script language for web-based multimodal
dialogue system. Figure 1 shows that in addition to a
client/server paradigm there is a vocal server between
the client and the web server. The vocal server is a speech
engine which embeds the ASR (Automatic Speech
Recognition) and the TTS (Text-To-Speech) technology.
The ASR engine recognizes user speech submitted from
the client. The TTS engine synthesizes textual prompt
provided from web server and returns speech output to the
client. In this architecture the vocal part of the multimodal
interaction is accomplished from the vocal server.

![Figure 1: SALT based system’s architecture](image)

The second solution uses the traditional client/server
architecture in the same way of visual web applications, as
shown in . In this case the ASR and TTS technology are
embedded together in the browser, which is a Multimodal
Browser. This multimodal browser reads XHTML+Voice
pages. XHTML+Voice is a language for multimodal
interfaces.

This language combines the visual language XHTML,
HTML enced to XML, with the vocal language
VoiceXML. Both the vocal and the visual interaction are
accomplished from the browser in the client side.

THE PROPOSED SYSTEM

The system Architecture

The application has been developed using the multimodal
technology VoiceXML based. We have chosen this solution
for two reasons. The first reason is that VoiceXML is a
recognized W3C standard, XHTML+Voice language is
under standardization while the XHTML+SALT language
is not a standard. The second reason is that in this architecture
the vocal technologies are embedded in the multimodal
browser, avoiding the need of a vocal server. This solution
enables every Web Server to provide multimodal application
because required vocal engine are provided with the
multimodal browser.

![Figure 2: VoiceXML based system’s architecture](image)

As shown in the system is based on a client-server
paradigm. It is accessible from small handheld devices like
cell phones and PDAs that serve many functions and contain
sufficient processing power to handle a variety of tasks. On
the client side the X+V application runs on the handheld
devices through the multimodal browser. The multimodal
browser is a software application that enables users to
display and interact via wireless connection with X+V
documents hosted by web servers. A detailed description
of X+V language and multimodal browser is in the next
subparagraph.

In the Figure 3 modules and information flow on the server
side are shown.

![Figure 3: The system information flow on the server side](image)

The multimodal application and the chat-bot module run on
the web server. Multimodal application files are generated
on demand from php document hosted by the web server.

Implementation technologies

The multimodal interface system has been developed using
the XHTML+Voice as implementation language, the
Multimodal toolkit as environment and the Multimodal
Browser as X+V pages interpreter.

XHTML+Voice language

The XHTML+Voice language, called X+V for short,
submitted to W3C for standard, is an XML based language
that ises up from combination between XHTML (XML
extension of HTML) and VoiceXML (XML based language
for pure vocal interface implementation). An X+V web page
is a normal XHTML page with additional Voice forms
(VoiceXML 2.0 subset, McGlashan et al., 2004) in the heading. These voice forms cover both speech recognition and text to speech mark-up. The control of interaction between vocal and visual part is made through XML Event, so that voice handlers can be invoked through a standard EventListener interface. An event can occur when the page was loaded, the user clicks on a button, or he says something the speech recognizer doesn't understand. The consequence can be a speech message or the starting of a dialog with the user. The voice forms are a subset of VoiceXML form, a W3C standard language for pure vocal interface development. It is a mark-up language which controls the vocal conversation flow and sets the context with the use of grammar file. The context is the set of utterances user can say. These user legal utterances are defined in the grammar file, written in XML Form.

The Multimodal Developing Environment

The multimodal interface has been developed with the Multimodal Tools 4.1.2.2 for WebSphere Studio V5.1.2., available at IBM developerWorks site with IBM Scholar Program License,. The Multimodal Tools includes the IBM WebSphere Multimodal Toolkit and IBM Multimodal Browser for PC. The multimodal toolkit is a plug-in of WebSphere Studio which includes a developing, debugging and testing environment for multimodal application. At present the plug-in is available for Rational Web Developer and in alpha version for Eclipse which is a well known open source environment.

Multimodal Browsers

Two versions of multimodal browser for PDA are also available at IBM site: they have been developed in a strategic relationship with Opera Software (based on the Opera Browser V7.55) and ACCESS Systems Company (based on the NetFront Browser V3.1 by ACCESS Systems). Each one of them has been improved with proper extensions that include the IBM ViaVoice speech recognition and text-to-speech technology, allowing the user to interact on handheld device with multimodal applications built using X+V. Multimodal Browsers looks like traditional browsers except for a button with a microphone icon in the toolbar, which let have vocal input control.

A.L.I.C.E.

A.L.I.C.E. (Artificial Linguistic Internet Computer Entity) is an artificial intelligence natural language chat robot based on an experiment specified by Alan M. Turing in 1950. A.L.I.C.E. has won three times (2000, 2001, 2004) the annual Loebner Prize which declares to “advance AI and serve as a tool to measure the state of the art”, which makes it a good candidate tool for human-like natural-language based dialogue systems. The A.L.I.C.E chat-bot's knowledge base is composed of question-answer modules, called categories and structured with AIML (Artificial Intelligence Mark-up Language), an XML-like language designed for creating stimulus-response chat robots. The question, or stimulus, is called the “pattern” while the answer, or response, is called the “template”. The template is composed of a natural language sentence and sometimes by other tag AIML, that can transform the response in a software that can save data, can start other programs, can give conditional answers and can recursively call other categories.

To achieve efficient pattern matching time, and a compact memory representation, the AIML software stores all of the categories in a tree managed by an object called the Graphmaster. The graph branches are the words representing the cultural baggage of the chat-bot, the path from the root to a terminal node represent a specific pattern and the terminal node links the relative template.

The Graphmaster merges patterns with common prefixes in the same pathways, achieving considerable compression. The dialogue mechanism is based on a search algorithm that is a special case of backtracking, depth-first search. The dialogue algorithm looks for a match between the user’s question and the chat-bot patterns (pattern matching).

ALICE-KB INDUCED GRAMMAR

To carry out a conversation with the chat-bot a set of AIML files concerning the specific domain has been written. The AIML categories are stored in the Graphmaster as shown in Figure 4. From this structure it is easy to induce an XML grammar form. This grammar is used by the multimodal system to understand the spoken user utterance.

![Figure 4: Alice’s Knowledge Base Tree](image)

To this purpose the list of utterances that the user can pronounce during the interaction with the chat-bot has been extracted from the AIML files. A parser has been created to translate this list of utterances into an XML Form grammar file as follows:

```xml
<one-of>
  <item> hello </item>
  <item> where </item>  
  <one-of>
    <item-repeat="0 -1"> is </item-repeat>
    <item-repeat="0 -1"> the </item-repeat>
    <item-repeat="0 -1"> talamone sculpture </item-repeat>
    <item-repeat="0 -1"> asclepius temple </item-repeat>
  </one-of>
</one-of>
</item>
</one-of>
</one-of>
```
The process is executed dynamically: the application runs a php file that runtime generates the grammar file allowing the system to adapt itself during the dialogue. If the number of AIML files is too large, it is convenient to generate the grammar offline, in order to reduce the response time of the system.

By means of the multimodal browser it is possible to recognize the user utterance and to synthesize the Chat-Bot textual reply.

The ASR (Automatic Speech Recognition) is obtained through grammar form. X+V can support the XML Form and the ABNF Form of the W3C Speech Recognition Grammar Specification (Andrew et and McGlashan, 2004). For this application we have chosen the XML Form.

THE SYSTEM IMPLEMENTATION: A DIALOGUE EXAMPLE SIMULATION

The system is accessible through a web page in a multimodal browser; this choice allows the reduction of the computational and memory requirements of the PDA.

When the user accesses the system from his handheld device, a multimodal browser is open. After a setup procedure where the user profile is settled, the multimodal browser asks the web server for the first X+V page and the dialogue interaction starts.

The interaction occurs through the loading of X+V pages; in every page the user can have a vocal dialogue. The multimodal browser, for any matched user utterance, loads the new page with the retrieved information and waits for a new vocal input.

Whenever the user pronounces his request, the multimodal browser looks for a match in the grammar file. If a match is found, and the user confirms it, the application answers to the request and the multimodal browser loads the new page, otherwise it asks the user to repeat the question.

Every time a X+V page is requested, the web server executes the correspondent php commands to generate it. In order to collect the information needed for the page creation, the application submits the recognized query to the chat-bot waiting for the answer.

The Chat-bot module looks for the best match among the rules in its knowledge base and sends a related query to the well-known search engine google in order to find further related information. The chat-bot answer is embedded by the application in a X+V page which is then displayed in the PDA Multimodal Browser as shown in .

An example of interaction is shown as follows:

**User:** Talk about the foundation of Agrigento.

**Chat-bot:** The foundation of Agrigento concludes the vast colonizing movement from Greece towards the West. According to the historical tradition. The city was founded, around 582 B.C., by a group of settlers from Gela.

**User:** I need information about Agrigento valley.

**Chat-bot:** The main temples at the Acragass's valley are Concordia, Castor and Pollux, Hera, Hercules, Vulcan or Hephaistos, Zeus Jupiter.

**User:** Tell me about Concordia temple at Agrigento.

**Chat-bot:** Erected around 430 B.C., it is 19,758 by 42,230 meters, slightly larger than a double square. It covers an area of 843.38 square meters and has a height of 13,481 meters. The cella was preceded by a simple antechamber with two columns and was followed by a back porch, where the treasure, votive gifts and the archives of the temple were kept.

![Figure 5: An example of interaction](image)

CONCLUSION

A multimodal information retrieval system has been presented to assist user of a cultural heritage in visiting historical sites in a natural manner overcoming the interaction limitation of traditional pre-recorded audio/visual guides.

The system simulation has shown the advantages of chat-bot and speech recognition technologies embedded in a Personal Digital Assistant, which can be easily carried during the site tour. Future work will regard the enhancement of the system improving the interaction naturalness.

REFERENCES


Alice: [http://www.alicebot.org](http://www.alicebot.org)


Microsoft Multimodal Technology http://www.microsoft.com/technet/prodtechnol/speech/library/ msos/sas_are_ev.mspx


AUTHOR BIOGRAPHY

AGNESE AUGELLO received her “cum laude” Laurea degree in “Ingegneria Informatica” from the University of Palermo on April, 2004. She started her Ph.D. in the “Computer Science and Artificial Intelligence” Lab of DINFO Dipartimento di Ingegneria Informatica, University of Palermo, Italy. Her research interests are in the field of Natural Language Processing and Human-Computer Interfaces.

ANTONELLA SANTANGELO received her “cum laude” Laurea degree in “Ingegneria Informatica” from the University of Palermo on July, 2005. She started her Ph.D. in the “Computer Science and Artificial Intelligence” Lab of DINFO Dipartimento di Ingegneria Informatica, University of Palermo, Italy. Her research interests are in the field of Natural Language Processing and Multimodal Human-Computer Interfaces.

ANTONIO GENTILE is associated professor at the DINFO Dipartimento di Ingegneria Informatica of the University of Palermo. His research interests are currently digitel architecture and natural language process. He is a member of the IEEE, IEEE Computer and ACM technical society.

GIOVANNI PILATO received his “cum laude” degree in “Ingegneria Elettronica” and the Ph.D. degree in “Ingegneria Elettronica, Informatica e delle Telecomunicazioni” from the University of Palermo, Italy, in 1997 and 2001, respectively. Since 2001 he is a staff research scientist of the ICAR-CNR. He is also Lecturer at the DINFO Dipartimento di Ingegneria Informatica of the University of Palermo. His research interests include geometric techniques for data mining and natural language processing.

SALVATORE GAGLIO is full professor of Computer Science and Artificial Intelligence at the DINFO Dipartimento di Ingegneria Informatica, University of Palermo, Italy. He is member of various committees for projects of national interest in Italy and he is referee of various scientific congresses and journals. His present research activities are in the area of artificial intelligence and robotics. Currently he is also Director of the Palermo branch of ICAR-CNR.
ORGANIZATIONAL SIMULATION
IMPROVING A CALL CENTER PERFORMANCE AT A TELECOMMUNICATION COMPANY

Fawaz Abdulmalek
Kuwait University
Department of Industrial and Management Systems Engineering,
College of Engineering and Petroleum
Kuwait University, P.O. Box 5969, Safat, Kuwait
E-mail: Abdulmalek@kuriv.edu

KEYWORDS
Simulation, Telecommunication, Forecasting.

ABSTRACT

One of the problems faced by a mobile telecommunication company in Kuwait is the excessive waiting time to handle phone calls in its call center. First a Moving Average forecasting model was developed to scientifically forecast the call arrivals to the call center. The Moving average forecast proved to track call arrivals better than the current reactive forecast. Second, based on the Moving Average forecast an improved agents schedule was developed. Finally, to test the effect of the Moving Average forecast and the new agent schedule on the customer waiting time, a simulation model was developed, validated, and employed. The simulation results indicated that the new agent’s schedule using the Moving average forecast could reduce the average customer waiting time potentially compared to the current agent’s schedule using the actual call arrivals.

INTRODUCTION

Many companies seek the means to compete with their competitors by providing superior customer service. One industry where the competition is fierce is the call center industry, in particular, the mobile telecommunication industry. The application of operations research modeling (queuing, simulation, linear programming, …) to the call center industry is receiving significant attention recently, see, for example, (Duder and Rosenwein 2001) and (Mehrotra 1997 ). It is important to note that the cost of providing trained agents on-line to answer customer enquiries accounts for over 50% of total operations costs, (Duder and Rosenwein 2001). This is because it has been recognized that a critical factor in business success is being able to respond quickly to customer requests, (Whitt 1999). Hence, the costs of staffing telephone call centers have become a significant part of business expense. Therefore, many companies have used operations research modeling to efficiently staff their call centers, (Andrews and Persons 1989), (Andrews and Persons 1991), (Brigandi et al. 1994), (Lin et al. 2000) , (Cezik et al. 2001), (Tych et al. 2002).

In administering a call center of a mobile telecommunication company in Kuwait, the management would like to scientifically forecast the call arrival rates in order to have better agent’s schedule. The call center provides services to customers calling by phone twenty four hours a day seven days a week. These services include billing inquiries and payments (Automatic and talking to an agent), invoices, maintenance, and other general information about the company. The company seeks ways to increase its market share and to improve its services. The goal of the call center is customer satisfaction, meaning that an incoming call must be answered by an agent with the minimum delay time possible. It has been found that currently a customer waits, on average, three minutes. This waiting time is considered excessive for a call center which must provide service as quick as possible to customers. Thus, it is important to have a good agent’s schedule according to call arrivals in order to minimize the cost of waiting time.

The problem is a queuing problem where there is a single queue and servers (number of agents) in parallel. After collecting an extensive amount of data for both interarrival times and service times, we found that neither interarrival times nor service times follow exponential distribution, and hence, the commonly-accepted queuing model of M/M/s may not be appropriate. Therefore, simulation was used to model the problem. First a Moving Average forecasting model will be developes to scientifically forecast the call arrivals into the call center. Second, an agent’s schedule will be employed based on the Moving Average forecast. Finally, to test the effect of the Moving Average forecast and the new agent’s schedule on the customer waiting time, a simulation model will be developed, validated, and employed.

PROBLEM DESCRIPTION

The call center is one of the customer care sub departments which comes under the sales group in the company. The call center serves customer calling by phone. The call center provides the following services:

- Subscription information, billing inquiries and payments.
- Attendant help for requiring additional information.
- Technical and maintenance emergencies
- Contract and account details.
- Special services (e.g., changing pin codes)
These services have to be performed in the best possible way in order to satisfy customers and enhance the company's reputation because of one major competitor in the telecommunication market in Kuwait. With this in mind, the goal of the call center is whenever a customer calls, an agent answers with the customer waiting for a reasonable amount of time. Currently the company is having staffing problems due to the lacking of a sound scientific forecasting method to predict call arrival patterns. The customer currently waits an average of three minutes before being handled by an agent. This waiting time is considered very high and intolerable by customers. Next, a forecasting model will be developed in order to predict the actual call arrival rates so as to have a better agent's scheduling.

**FORECASTING**

Currently the company would like to predict how well the call center is staffed for a given day so as to efficiently schedule agents. Past six months data was provided for every half hour period of the day. Currently the call center schedules agents on the fly and no scientific method is used. The data for the actual number of call arrivals were plotted, and a random pattern was observed. A moving average forecasting method (MA) with span of four (α = 4) was found to perform well with small mean absolute forecast error (MAFE). So it was used to forecast the call arrivals per half hour period.

MA was applied to a randomly chosen days of the week and it was found that overall the method tracks call arrivals closer to the actual calls than the method used by the company does.

**AGENT SCHEDULING**

In order to have a good agent’s schedule, it is critical first to find if there exists any difference of call arrival patterns throughout the day among different days of the week for every half hour period. The average number of calls for each day was obtained. This average represents the average of all call arrivals per half hour period for a given day, totaling 48 periods per day for the past six months. It was observed that there is no significant different between the weekdays and they all follow the same trend. On the other hand, Thursday and Friday (weekends) appear to be different for some periods. To test this scientifically, hypothesis testing will be performed next.

**WEEKDAYS HYPOTHESES TEST**

First, a hypothesis test is applied to test whether there is a significant difference of call arrivals per period between weekdays. For a significant level (α) of 0.05, twelve periods were randomly chosen (e.g., 00:00 and 20:00) and the hypothesis is as follows:

- **H₀**:  $\mu_{Sat} = \mu_{Sun} = \mu_{Mon} = \mu_{Tues} = \mu_{Wed}$
- **H₁**: At least two are different

All P-Values for the period chosen are larger than α, for example P-Value of 0.9668 was obtained for period 00:00. This indicates that there is no significant difference between call arrivals per period between the weekdays.

**WEEKEND HYPOTHESIS TEST**

Second, hypothesis testing to test whether there is a significant difference of call arrivals per period between weekdays and weekends is performed. The test is carried out for periods 09:00, 11:00, and 15:00 which were arbitrary chosen. The test is carried out for $\alpha = 0.05$.

**WEEKDAYS AND THURSDAY HYPOTHESIS TESTING**

To test whether there is a significant difference of call arrivals per period between weekdays and Thursday the following hypothesis is used:

- **H₀**: $\mu_{Weekdays} = \mu_{Thursday}$
- **H₁**: $\mu_{Weekdays} \neq \mu_{Thursday}$

In both periods 09:00 and 12:00 the P-values (0.000 and 0.003) were found to be less than α indicating a significant difference for those periods, however, in period 15:00 the P-value is 0.063 which is greater than α which indicates no different for this period. A test with α of 0.1 on the other hand concludes that there is a significant difference of call arrivals per period even for period 15:00 between Thursday and weekdays.

**WEEKDAYS AND FRIDAY HYPOTHESIS TESTING**

To test whether there is a significant difference of call arrivals per period between weekdays and Friday the following test hypotheses is used:

- **H₀**: $\mu_{Weekdays} = \mu_{Friday}$
- **H₁**: $\mu_{Weekdays} \neq \mu_{Friday}$

In both periods 09:00 and 12:00 the P-values (0.000 and 0.000) were found to be less than α indicating a significant difference for those periods, however, in period 15:00 the P-value is 0.964 which is greater than α which indicates no different for this period. A test with α of 0.1 supports the above.

In conclusion there is a significant difference of call arrivals per period between Thursday and weekdays at an alpha of 0.1, whereas there is no significant difference of call arrivals between Friday and weekdays.

**THE IMPROVED AGENT'S SCHEDULE**

An improvement is made to the current call center agent's schedule. Currently the call center creates their agent's (teams) schedule reactive to their forecast where each team works for an eight hour shift with one hour break for each agent. It was found that in some periods call rate is high while the assigned
agents are inadequate whereas in other periods more agents are assigned than needed. To better allocate agents to working periods, a new improved schedule was created using MA forecast by simply assigning the maximum possible number of agents at periods with high incoming call rates, and placing the agent’s breaks at periods with low incoming call rates taking into account the break which can be segmented in 15 minutes increment. By doing this, one minimizes the call rejection rate and maximizes agent’s utilization

**SIMULATION MODEL**

In order to implement the new agent’s schedule, a simulation model was developed to test the effect of the new agent’s schedule on customer waiting time. The simulation was developed using Arena package. Data including agent’s service times and call arrival rates were provided by the company. The data used in this model is from September 2003 until April 2004. All the statistical distributions used in the simulation for call Interarrival rates, service times, delay times, and others were determined by using the Input Analyzer of Arena. The Input Analyzer uses three types of goodness of fit tests. Based on these tests, the best fit was chosen. Hypothesis testing was applied to the service time of twenty agents in three different shifts in order to test if there is any difference exists between agent’s service times for the different services provided (e.g., balance inquiry or payment). With alpha of 0.05 the test is as follows

\[ H_0: \mu_1 = \mu_2 = \mu_3 = \mu_4 = \mu_5 = \mu_6 = \mu_7 = \mu_8 = \mu_9 = \mu_{10} = \mu_{11} = \mu_{12} = \mu_{13} = \mu_{14} = \mu_{15} = \mu_{16} = \mu_{17} = \mu_{18} = \mu_{19} = \mu_{20} \]

\[ H_1: \text{At least two are different} \]

Since the P-value = 0.9997 is greater than \( a \), agent service times are the same for all services. The service time distribution was found to be lognormal (2.49, 3.7).

The model starts when calls are transferred from the interactive voice response system (IVR) to the call center. An arriving call will go to a decide module to be accepted on average, 86% of the time or abandoned 14% of the time. An accepted call will then be handled by an agent according to a schedule which differs every 30 minutes defined in the schedule module. A sub-model was constructed to keep track of the current time period during a day (48 total periods). After being serviced, the customer exits the system. An abandoned call will directly leave the system without being answered.

**Simulation Verification and Validation**

Verification is the process that assures that the simulation model copies the real system, (Law and Kelton 1999). A cautious trace study was followed by tracing an entity once it is created until it is disposed from the system along with detailed animation and it was verified that the model adequately replicated the real system.

Validity of the model was carried out by comparing the output of the model to those from the real system by using the current call center agent’s schedule. A comparison was done between the actual average service time, the average waiting time, the percentage of answered calls, and the percentage of abounded calls with those from the simulation results. Table 1 presents tie results where it is apparent that data from the real system validates the results of the simulation model.

<table>
<thead>
<tr>
<th></th>
<th>Actual Results</th>
<th>Model Results</th>
</tr>
</thead>
<tbody>
<tr>
<td>Service time</td>
<td>2.39 min</td>
<td>2.49 min</td>
</tr>
<tr>
<td>Waiting time</td>
<td>2.73 min</td>
<td>2.55 min</td>
</tr>
<tr>
<td>Answered Calls</td>
<td>86 %</td>
<td>85.6%</td>
</tr>
<tr>
<td>Abounded Calls</td>
<td>14 %</td>
<td>14.37 %</td>
</tr>
</tbody>
</table>

**SIMULATION EXPERIMENTS**

To analyze the situation on hand, five different scenarios were tested using the simulation model and they are as follows:

1- Current agent’s schedule, using current forecast for call arrivals.
2- Current agent’s schedule, using the actual call arrivals.
3- Improved agent’s schedule, using the current forecast for call arrivals.
4- Improved agent’s schedule, using the actual call arrivals.
5- Improved agent’s schedule, using MA forecast for call arrivals.

For all scenarios, Simulation was run for an 8 hour-shit three shifts a day for 300 days (replications). Three hundred replications were considered good enough according to criteria described by (Law and Kelton 1999).

**SIMULATION RESULTS**

Here the simulation result for Saturday (3/4/2004) is arbitrary chosen to present the results. As mentioned previously the performance measure used here is the average customer waiting time. Table 2 shows the average waiting times for the Five scenarios.

As shown in the table, the average waiting time for the improved schedule using the current forecast (Scenario 3) is smaller than the average waiting time for the current schedule using the current forecast (Scenario 1), a reduction of approximately 93%. Also scenario 4 average waiting time is smaller than scenario 2 average waiting time, an reduction of almost 81%. Scenario 5 which is for the improved schedule using MA average forecast provides an almost 89% reduction in the average waiting time when its compared with scenario 2. Also when scenario 5 is compared with scenario 4 which is for
the improved schedule using the actual calls arrival rate, a reduction of an almost 29% is obtained. In general, the results obtained for this day for the average waiting time was also validated for other randomly chosen days (including Thursday and Friday) and almost exactly the same results were attained.

Table 2. Average waiting times for the five scenarios

<table>
<thead>
<tr>
<th>Scenario</th>
<th>Schedule Type</th>
<th>Calls arrival type</th>
<th>Average waiting time (min)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Current</td>
<td>Current Forecasted</td>
<td>0.4885</td>
</tr>
<tr>
<td>2</td>
<td>Current</td>
<td>Actual Calls</td>
<td>0.7402</td>
</tr>
<tr>
<td>3</td>
<td>Improved</td>
<td>Current Forecasted</td>
<td>0.0349</td>
</tr>
<tr>
<td>4</td>
<td>Improved</td>
<td>Actual Calls</td>
<td>0.1373</td>
</tr>
<tr>
<td>5</td>
<td>Improved</td>
<td>MA Forecast</td>
<td>0.0972</td>
</tr>
</tbody>
</table>

CONCLUSION

The first task of this research was to come up with a forecasting scheme to better track staffing during the day for the call center. Moving Average forecast was applied and its was found that overall the method tracks call arrivals closer to the actual data than the current method used. Second, a new agent’s schedule was created using MA forecast. The current schedule the call center uses is just done on the fly reactive to the actual calls arrival rate. The new schedule staff teams in periods with high call arrival rates and places agent’s breaks with period of law incoming call arrival rates. Last, a simulation model was used to test the effect of the new agent’s schedule and the Moving Average forecast on the average customer waiting time. It was shown that the new agent’s schedule using the Moving average forecast could potentially reduce the average waiting time almost to 89% compared to the current agent’s schedule using the actual call arrivals.

REFERENCES


AUTHOR BIOGRAPHY

Fawaz Abdulmalek is a faculty member in the Industrial and Management Systems Engineering Department at Kuwait University. He received his B.S. degree in Mechanical Engineering from the University of South Carolina, and his M.S. and PhD degree in Industrial Engineering from the University of Pittsburgh. His areas of interest are Production and Inventory Control, Lean Manufacturing, Simulation and Supply Chain and Logistics. For further information please check: http://www.faculty.eng.kuniv.edu.kw/staff/fmast10.jsp.
A NEW HIGHLY FLEXIBLE SIMULATION MODEL FOR AN ADAPTABLE, EFFICIENT AND SAFE SECTORIZATION IN AN AIR TRAFFIC AREA CONTROL CENTER

Lorenzo Brunetta  
University of Padova  
Via Gradenigo 6/A  
35131 Padova PD  
Italy  
E-mail: brunetta@dei.unipd.it

Giorgio Romanin-Jacur  
University of Padova  
Stradella San Nicola 3  
36100 Vicenza VI  
Italy  
E-mail: romjac@dei.unipd.it

KEYWORDS  
Aerospace, Discrete simulation, Air Traffic Control, Area Control Center, Flexible sectorization.

ENVIRONMENT AND CURRENT LITERATURE

The air traffic control (ATC) system gives guidance to prevent collisions and manage efficient traffic flow. Each country manages its own airspace divided in control areas by means of Area Control Centers (ACCs); each ACC rules all aircrafts flying inside its Land Of Responsibility (LOR), serving both civil and military users; as an example, Italy has four ACCs: Milan, ruling NW area, Padova for NE, Rome for SW and major islands, Brindisi for SW; NE area is reported in Figure 1. Air traffic demand placed on the system depends on aircraft type, flight type, and possible landing and take-off operations. The airspace controlled by each ACC may be subdivided into vertical and horizontal “sectors”; each sector is characterized by its lower and upper height limits and by its border. For Padova ACC, at the beginning of the study, the airspace was divided into nine sectors, as reported in Figures 1 and 2; the area was horizontally divided into a North and a South part and vertically into five segments; consequently NTT and STT (North and South Top Top), NTS and STS (North and South Top Sectors), NUS and SUS (North and South Upper Sectors), NLS and SLS (North and South Lower Sectors) were defined; AD (Arrival and Departure) Sector includes a smaller area evidenced in Figure 1. Enroute air traffic controllers are persons in charge to expedite and maintain a safe and orderly aircraft flow over the sectors taken in charge through the use of long-range radar sets. In areas of high demand, it is necessary to impose restrictions to ensure that the number of flights within a sector does not exceed a safe limit. This figure is called “sector capacity”: that is, the number of flights that can be handled by air traffic controllers in a given sector, within a specified time, without infringing safety or causing overload to normal controller activities.

Today technologies allow a flexible “sectorization” of an ACC depending on the traffic demand. Controller workload is a function of the number of aircraft flying in the same sector and of safety constraints; this implies that there is an upper limit on the number of planes that can be simultaneously controlled. Different sectorizations may affect this limit.

The day-to-day problems faced by the air traffic control system are primarily related to the volume of air traffic demand placed on the system, and weather. In order to properly satisfy the constantly growing air traffic demand, there is the need of an “increased use of airspace capacity”, see ACARE (2001).

In the European context, technology has tremendously improved the quality of work of air traffic controllers, augmented the controller workload and, at the same time, assured the required safety standards. A better use of airspace capacity will certainly reduce system inefficiencies, increase system capacity, and will possibly reduce delays. Every ATC action falls into one of these three controller task categories: routine task, level change monitoring task, conflict monitoring task. Important factors influencing sector capacity are the task duration and the task occurrences (the number of routine, climb/descent, and conflict tasks). Recently, within the Eurocontrol COCA project, as reported by Flynn, Christian and Benkouar (2003) a quick and precise controller workload and sector capacity estimation methodology (macroscopic) was developed. However, an important factor has not been explicitly considered: Team Resource Management (TRM). TRM can be seen as a set of strategies for the best use of all available resources - information, equipment, and people - to optimise safety and efficiency of air traffic service.

A team in Air Traffic Management (ATM) is a group of two or more persons who interact dynamically and interdependently within assigned specific roles, functions and responsibilities. They have to adapt continuously to each other to ensure the establishment of a safe, orderly and expeditious flow of air traffic. Among controllers the smallest “teamwork” cell in ATM could be described as the controllers and flight data assistants working within the same sector. There is also teamwork between controllers of different operational areas (adjacent sectors both within the same ACC or belonging to different ACCs). Teamwork relations are also important between controllers and any other operational staff (flow management, supervisors, ATM support staff, etc.).

To our knowledge, only little work related to the issues described above is available in the scientific literature. Mathematical models used to assess sector capacity rely on estimating or predicting the controller’s workload and are based on measuring pre-defined, event driven, controller tasks of a specified duration. Tasks are described and grouped in different ways, as for example in Christien and
Benkouar (2003) or in Flynn, Christien and Benkouar (2003). Analytical models currently in use are able to provide a precise, but pessimistic assessment of sector capacity in an Area Control Center (ACC), see Flynn, Christien and Benkouar (2003). Given the significant technological improvements derived by new equipment and the recognized importance of TRM, the numbers produced by existing models are values that underestimate the real figures, thus resulting in an under-utilization of the air space and inefficiency. In particular those models do not recognize the fact that a controller sector is not isolated: as a result, sector capacity is defined independently of the adjacent sectors, while it is widely recognized that a proper coordination among adjacent sectors may lead to an overall capacity that is higher than the simple arithmetic sum of their capacities. In addition controller task durations are considered as static parameters. The only paper considering a dynamic model of ACC operations is the one by Brunetta and Romanin-Jacur (2005): there different sectorizations for the same airspace are simulated, and related controllers’ workload in all day time intervals are computed; the airspace subdivision is the one seen above and experiments are performed related to a well defined air traffic; some rigidity may be noted, as the air traffic behaviour during the day and the times when various sectors are activated or deactivated are set a priori, therefore for every aircraft the sectors to be crossed are exactly known.

PRESENT RESEARCH AND RESULTS

This research has been performed by a group of people with complementary expertise: researchers from Academia and air traffic controllers with TRM training working at the Padova ACC. As already seen in Figures 1 and 2, Padova ACC rules air traffic over North East Italy. North East Italy airspace is divided into five flight levels; the four upper ones are further subdivided into a North and a South sector; the first level presently includes an only sector dedicated to arrivals and departures (AD), extended on a reduced area with respect to the total LOR (the remaining area is included in the two sectors sharing the second level). During our study, such division was modified to look at possible improvements. In a first test, the current lowest level was extended to the whole LOR area and subdivided into two sectors, a North and a South Lowest sector; in a second test, a West Lowest Sector and an East Lowest Sector, evidenced in Figure 3, were suggested: the latter test gave the better results in terms of traffic workload, which appeared more balanced among various sectors and never overcame the sector capacity limits.

Division into sectors is made a priori, but sectorization may be chosen as a function of traffic demand, in that some or all sectors may be merged together during low demand periods, like for instance in the night, when an only large sector is active. With the aim of obtaining a good flexible sectorization for different situations, we performed two different studies; in the first study, we adapted the model presented by Brunetta and Romanin-Jacur (2005) in order to test different division of the airspace into sectors. We modified the already built MicroSaint discrete event simulation model with the objective of describing air traffic flows in the Padova LOR, in order to compare the existing sectorization with the two new ones described above, with related implications for controllers workload and overall air traffic safety. In both partially new models aircraft entrance times into sectors and stay times within sectors are random variables, whose mean and standard deviation depend on the sector, the flight type and the aircraft type. Models’ results report all sectors’ workload, both “instant workload”, which consists in the number of aircraft simultaneously present inside the sector as a function of time, and the hour workload, i.e., the total number of aircraft which crossed the sector hour by hour. We tested the two new sectorizations during an exceptionally busy day. After an accurate analysis of flying data from 01/01/2003 to 11/05/2004 we selected the most representative day, i.e., August 31st, 2003, where 2038 aircrafts flew within the LOR. The present sectorization (Figures 1 and 2) showed to generate controllers dramatic overload in the peak hours of the day, with consequent labour and pain to maintain the required safety standard. The first suggested new sectorization, consisting in the two new North and South sectors at the lowest level, including a larger area with respect to the old one, is sufficient to reduce and successfully balance all sectors’ workload; in particular we change the workload of controllers of the second flight level from 40 to 32 aircraft per hour in the North Lower Sector and from 38 to 21 aircraft per hour in the South Lower one. The second suggested sectorization, consisting in the two new West and East sectors at the lowest level, gave the same advantages plus a better equilibrium between the two lowest sectors’ workload. In the model the various sectors may be activated in different ways during the day, by merging some of them during low demand intervals, according to a predetermined schedule. The model is essential and therefore easily understood and used; some ability is required to change the adopted sectorization schedule, due to the rigidity evidenced in the previous paper section. The old model was dramatically and usefully improved for what concerns flexibility, by building up a second model, which totally separates aircraft generation and sector generation. In other words, one section of the model, described in detail in Figure 5, manages aircraft entering the LOR during the day, the other section manages the adopted sectorizations during different periods of the day, and consequent addressing of aircraft to activated sectors; an example of the model section managing 10 sector configuration is reported in Figure 6. Such a way different sectorization schedules, i.e., different sequences of changes of sector configuration during the day.
Figure 1: NW Italy area controlled by Padova ACC; sector borders are evidenced.

Figure 3: NW Italy area controlled by Padova ACC; new lowest level borders suggested are evidenced.

<table>
<thead>
<tr>
<th>NTT</th>
<th>STT</th>
</tr>
</thead>
<tbody>
<tr>
<td>NTS</td>
<td>STS</td>
</tr>
<tr>
<td>NUS</td>
<td>SUS</td>
</tr>
<tr>
<td>NLS</td>
<td>SLS</td>
</tr>
<tr>
<td>AD</td>
<td>Gnd</td>
</tr>
</tbody>
</table>

Figure 2: Sectorization of Padova ACC airspace (vertical section): lower and upper sector limits are evidenced.

Figure 4: Example of results: instant workload (simultaneously controlled aircraft) and hour workload (total controlled aircraft) for two different sectors.
may be quickly tested, corresponding to the same aircraft arrivals; alternatively, the same sectorization schedule may be tested, corresponding to different aircraft arrival behaviours, related to different traffic situations.

By such a second model many experiments may be performed in a short time, to evaluate operators’ workload consequent to different decisions about activated sectors, in correspondence to different traffic demand behaviours.

It is to be noted that the model may run on a PC and requires a low cost simulation software.

It is obvious that the model can be easily and usefully adapted for any other ACC and related LOR.

REFERENCES


MILITARY ORGANIZATIONAL SIMULATION
ABSTRACT

The US Army vision, announced in October of 1999, encompasses people, readiness, and transformation. The goal of the Army vision is to transition the entire Army into a force that is strategically responsive and dominant at every point of the spectrum of operations. The Future Combat Systems (FCS) program is a joint DARPA/Army program that is identifying the promising systems and technologies for achieving the Army's vision of fielding an "Objective Force" beginning this decade. The Objective Force is the Army's future full spectrum force; organized, manned, equipped and trained to be more strategically responsive, deployable, agile, versatile, lethal, survivable, and sustainable across the entire spectrum of military operations from major theater wars through counter terrorism to homeland security. This force will be lighter and more mobile and the Army transformation requirements include the ability to put a combat-capable brigade anywhere in the world within 96 hours, a full division in 120 hours, and five divisions on the ground within 30 days. The FCS is envisioned to be a versatile family of combat systems that represent the way the Army will fight wars in the future. The Army plans to field an integrated system of combat systems in blocks until full objective capability is realized to meet the goals of Army transformation. These systems will incorporate the latest technologies for mobility, lethality, sensor platforms, and survivability. The Tank Automotive Research, Development, and Engineering Center (TARDEC) has spearheaded these efforts by developing collaborative relationships with leading US universities to investigate and implement fostered research into the virtual prototyping process. The goal is to streamline and improve simulation-based design and analysis capabilities with special emphasis being placed on application to new and existing tactical wheeled and tracked vehicles. This paper will describe some of TARDEC's efforts to develop more robust, general purpose, analytical capabilities which will allow detailed investigation of a vehicle's synergistic dynamic performance.

INTRODUCTION

The FCS "system of systems" designs as illustrated in Figure 1 will be accomplished primarily by using modeling, simulation and experimentation. The goal of the FCS program is to design an ensemble that strikes an optimum balance between critical performance factors, including ground platform strategic, operational and tactical mobility; lethality; survivability; and sustainability.

Through university-based research consortiums, the Tank Automotive Research, Development, and Engineering Center (TARDEC) is researching and developing modeling and simulation tools that will be necessary to demonstrate the feasibility and operational potential of advanced commercial and military technologies. TARDEC’s goals are to have the necessary M&S tools in place to investigate, demonstrate, and mature new and improved combat vehicle and automotive technologies to enable transformation of the Army to the Objective Force. The objective of this paper will be to describe several TARDEC-sponsored, university-based research programs in areas such as Dynamics and Control of Vehicles and Mobile Robots, Human Centered Design and Simulation, High Performance Structures and Materials, and Advanced and Hybrids Powertrains that build on the Army's simulation investments into tools that will be utilized to investigate and answer FCS real-world vehicle design, acquisition, and engineering support questions. M&S research efforts are constantly striving to make the Army a smarter and more cost-effective buyer of equipment, and more importantly, significantly reducing the associated risks that are inherent in procuring newly designed, untested equipment. Due to the TARDEC’s recognized modeling and simulation focus and expertise, research and development projects to develop innovative and cost effective alternatives and meet the demands for improved...
fidelity and lower cost vehicle simulations, simulators, processes, and facilities have progressed and will be discussed.

DISCUSSION

FCS will, over time, replace the current fleet of 'heavy' vehicles - the Abrams tanks and Bradley Fighting Vehicles with a new family of manned and unmanned ground vehicles and aerial vehicles. The new manned ground vehicles will be both lighter and smaller than the current fleet. These lighter, smaller vehicles are designed to fit into a C-130-like plane, which allows them to be flown to a conflict anywhere in the world in 96 hours, rolled off and ready to fight. Part of what makes FCS transformational is its adherence to the new DOD acquisition model of spiral development, which allows developers to insert emerging technology as the systems mature over time. Also, the ability to interface with other military services, governmental agencies and multinational partners has been built into the FCS network from the ground up, making the system more relevant to regional combatant commanders.

In order to make these goals a reality, the vehicle design and procurement community has an urgent need for access to general-purpose, high-resolution, computer-based vehicle modeling and simulation methodologies suitable for rapid evaluation of dynamic performance characteristics, reliability, human operator performance, survivability, and maintainability. To reduce timelines and costs, Government engineering teams are increasingly being asked to develop and apply engineering-fidelity simulations to meet customer demands for simulation-based evaluation of vehicle designs, technology, and proposed product improvements throughout a vehicle's life cycle. With the rapidly changing landscape of national security threats, innovative future tactical and combat vehicle systems must be defined, created, and deployed faster than ever before. Yet, these new systems, which by definition will be more complex, must exhibit extremely high levels of reliability, durability, agility, and efficiency.

THE AUTOMOTIVE RESEARCH CENTER – HYBRID AND DIESEL ENGINE RESEARCH

The Automotive Research Center (ARC) is a US Army Center of Excellence in Modeling and Simulation headed up by the University of Michigan. This university-based center has become known as the “ARC” and works in partnership with seven other universities: Clemson University, Oakland University, the University of Alaska-Fairbanks, the University of Iowa, the University of Tennessee, the University of Wisconsin, and Wayne State University to fulfill the objectives of the ARC. The ARC has become a key basic research partner and the goal of this program has been to conduct basic research that advances the state-of-the-art in high fidelity modeling, simulation, and testing of military ground vehicles and provide linkage between automotive suppliers, manufacturers, and the military in the critical product development phase. University research projects must have Government and Industry participation and focused into basic research thrust areas. One of the ARC’s major research areas is in hybrid vehicles. The Army spends a large chunk of its budget on fuel for its vehicles and even a small increase in fuel economy would be paramount in savings. Significant future breakthroughs regarding fuel economy of heavy vehicles will require multiple new technologies in addition to advanced fuel converters, the most promising one being the hybridization of the propulsion system. The ARC is funding a project called ‘Optimal Control and Design of Hybrid Vehicles’ which is working to i) identify key issues in the abstraction of hybrid vehicle components, including fuel cells, ICE, electric drives, batteries and auxiliary components (for both performance and efficiency); and ii) to study the simultaneous optimization of hybrid vehicles, which involves the identification of its optimal component sizes and optimal control algorithm (see Figure 2).

Figure 2: Hybrid Propulsion System Components.

Activity leading up to this project included partnership with Industry counterparts on development of optimal control for a delivery truck. Another ARC project is looking at ‘Hybrid Propulsion System Components and Integration’ which will develop methodologies, models and simulations of hybrid propulsion systems required for maximizing fuel economy benefits over military and dual use truck missions as shown in Figure 3.

Figure 3: Analytical Model of a Hybridized Engine.
It will utilize the newly developed simulation tools to study matching, optimal design and power management of novel hybrid systems. Studies of this project directly supported development and demonstration of the hydraulic hybrid system for a medium-sized truck. A third major ARC project concerning M&S of hybrid vehicle is ‘Hybrid Propulsion Options’ which explores the benefits of hybrid propulsion systems depend on a combination of factors, such as architecture, type and design of components and power management. Among the critical decisions leading to the final design, selection of the vehicle configuration is perhaps the least understood. While the literature often refers to main types of hybrid architectures as parallel, series and power-split, the actual number of variants is much larger, including flexible concepts called parallel/series. Due to the increased complexity of proposed systems, it is impossible to rely on intuition and assessments of configurations based on predetermined component sizing and generic power management strategies. Therefore, this project is developing a simulation based methodology for systematic evaluation of hybrid propulsion options for selected classes of military and dual use trucks. The goal is to maximize the potential of suggested propulsion concepts to improve fuel economy and mobility. In addition, truck missions, and the impact of alternative duty cycles on the decisions regarding hybrid propulsion options are being investigated.

The ARC is also conducting research into the development of predictive, physics-based models capable of addressing details of diesel engine processes and enabling studies of novel designs and modes of operation’. One project, entitled ‘Diesel Spray, Combustion and Emissions’ is working to develop and validate a diesel engine simulation model which has the capability of predicting fuel spray evolution, ignition delay, combustion and exhaust emissions as depicted in Figure 4.

![Figure 4: Analytical Diesel Engine Model w/ Test Cell.](image)

The program consists of two major tasks. The first task is to develop a model for spray development and behavior in swirling air, the fuel distribution in the combustion chamber, the start of the auto ignition process, and the flame development and engine-out emissions. The second task is to verify the model on a small bore high speed, direct injection, diesel engine equipped with a common rail injection system. Another project entitled ‘Diesel Cold-Start Transients’ will utilize these analytical models to investigate the cold start characteristics of diesel engines, particularly under military operating conditions. The work includes detailed analysis of the thermodynamic, heat and mass transfer, auto ignition and combustion, and the development of mathematical models for the cranking period, and combustion instability. The models will be validated on a four cylinder 1.5 liter, turbocharged, intercooled diesel engine. The engine is equipped with a common rail injection system and a variable geometry turbocharger. The experimental work will be conducted in the cold room at different ambient temperatures. The outcome of this project is to develop a strategy for the electronic controls that would improve the operation of the engine under the severe environmental conditions.

**OHIO STATE UNIVERSITY – FUEL CELL RESEARCH**

Ohio State University (OSU) engineers and researchers have begun work to develop and implement new modules and methodologies aimed at driving faster innovation of fuel cell vehicle systems while mitigating risks. Newly developed and implemented modeling and simulation and virtual prototyping techniques are being used to demonstrate the feasibility and operational potential of advanced commercial and military fuel cell vehicle technologies with application to new and existing vehicle systems. They will also be used to describe the capabilities of potential future vehicles and investigate critical technologies that will be required to achieve future vehicle performance goals and objectives. This research will investigate various simulation-based, virtual prototyping concepts developed in commercial vehicle design that can streamline future vehicle system development processes. The FCS program does not focus on the development but on the integration of technologies and specifically, is looking at transitioning from traditional to future propulsion and on-board power systems for its logistics vehicles. OSU is investigating power and propulsion technologies in the commercial sector as they evolve from traditional combustion engine and mechanical transmission vehicle propulsion systems, to combustion engine hybrid-electric systems, to systems with on-board fuel cell electric power generation, to the use of fuel cell systems exclusively for propulsion. In short, the Army needs effective system analysis tools that can be employed to assess and optimize the use of fuel cell propulsion and on-board vehicle power technologies and platform design. Current modeling tools are not sufficiently mature to assist the Army to make the best acquisition decisions.

OSU researchers are working on a project called ‘System analysis and modeling tools for fuel cell traction and APUs systems’ which is focused on developing dynamic models as shown in Figure 5 of a pressurized Proton Exchange Membrane (PEM) fuel cell system and a Solid Oxide Fuel Cell (SOFC) stack and system model. The research will provide modeling, analysis, and simulation tools to evaluate PEM fuel cell performance during high bandwidth loads typical in real world conditions. The scope of this research encompasses development and
validation of a comprehensive PEM fuel cell model with water and thermal management capability for the application of the model to extreme environmental conditions.

Figure 5: Dynamic Model of PEM Fuel Cell System and SOFC Stack.

Enhanced capabilities of the models will allow investigation of the potential benefits of military applications of fuel cells as Auxiliary Power Units. Integration will be pursued in order to study trade-offs associated with fuel cell cooling system design and powertrain electrification concepts. Development of the analytical and computational methods will be utilized for fuel cell design and optimization, simulation based acquisition; experimentally validate electric traction, energy storage systems, and fuel cell energy conversion. The research will also be used to develop analytical tools for gathering on and off-road driving data and duty cycles that can be used to refine vehicle evaluation procedures; develop durability models and metrics; and develop fuel cell and electric drivetrain technology. The Fuel Cell Control Laboratory 4 (Figure 6) is designed to enable the implementation of multivariable controllers, fault detection, and diagnostic algorithms for the regulation of reactant flow and pressure, stack temperature, and membrane humidity.

Figure 6: OSU Fuel Cell Laboratory.

The laboratory’s primary activity is the development and testing of real time control and diagnostic systems to accelerate the use of fuel cells by enhancing their safety, increasing their efficiency, and validating their productive effectiveness in real-world applications.

UNIVERSITY OF IOWA – HUMAN MODELING RESEARCH

Army researchers are currently working with the University of Iowa (UI) to investigate the combined application of motion capture technology and Digital Human tools; including simulation and ergonomic analysis in the evaluation of the human interface for 3D conceptual weapon systems in virtual environments. These tools have been tested with implementations in vehicle dynamics, soldier-in-the-loop simulations, mechanism design, and robotics. Digital humans are biomechanically accurate models of humans of various sizes that exist in virtual environments. Digital humans are relatively a new technology, within the last 5 years becoming commercially available that enable the ergonomic analysis of completely digital conceptual systems, reducing if not eliminating the need to build expensive and time-consuming full-scale prototypes. Digital humans can be used to tell designers what they can see and reach, how comfortable they are, whether they can access controls, and other important information. This information helps design more effective tools, higher performance devices and machines, faster and for less cost. CCAD is involved in research concerning digital humans. The acceptance of motion capture technology as applied toward the animation of digital humans by the entertainment industry, for major movies and kid shows, is steadily increasing as both hardware and software mature. Taking advantage of the accuracy, repeatability, and ease of use for motion capture systems, ergonomic studies can now be developed for 3D digital concepts, which are based on actual human data, in lieu of the manual scripting methods currently.

Virtual reality immersive environments are relatively new technologies that enable human interaction with 3-Dimensional visualization of objects and products before they are manufactured. This technology differs from the digital human approach in that real humans are placed in a virtual environment and allowed to interact with digital models as if they were real.

Figure 7: Human Digital Models.

One focus of the research being conducted at CCAD is to determine what the advantages and disadvantages are of using digital humans and/or virtual environments. The goal is to develop design methods that will reduce the number of one-of-a-kind prototypes of systems that must be built before designs are finalized. This process appears to require both virtual reality technology and digital
humans. Digital Humans and Virtual Reality will assist the Army in evaluating vehicle performance and maintenance procedures, reducing logistics support, designing at lower costs and reduced lead times, evaluating combat effectiveness, conduct training of soldiers, and evaluating and design soldier-machine interfaces.

The University of Iowa is also conducting research into Uncertainty and Reliability-Based Design Optimization. A team of researchers from various organizations is currently working together to develop methodologies to assess the effects of uncertainties and optimized product design with reliability. These methodologies will be applied for physics-of-failure analysis of both mechanical and electronic packages in various military vehicle systems. In characterizing uncertainty, it is necessary to consider physical, statistical, and model uncertainties as input. Methodologies must then be developed to assess the impact of these input uncertainties to the output performance of the system. For example, as has been presented in the US Army Trailer fatigue life application, when the system design is deterministically optimized the optimized design is approximately 50% reliable. To assure the optimized design is also highly reliable, RBDO methods must be applied. This will lead 6-sigma designs for manufacturing.

![Figure 8: Characterizing Uncertainty.](image)

In the near future, these modeling and simulation methodologies, tools, and processes will be developed and applied to various Army and commercial vehicle problems to improve and assess reliability and to assist in weight reduction initiatives.

CONCLUSIONS

The major performance areas which are impacted most by high resolution vehicle simulations are mobility, stability, reliability and safety. Ground vehicles operate in very harsh environments and are expected to perform their intended missions. Vehicle developers are responsible for setting realistic performance specifications and ensuring that they are met. MS is making the Army a smarter and more cost-effective buyer and tester of vehicles and equipment, and more importantly, significantly reducing the associated risks to personnel and property that are inherent in a war-fighting environment. In addition, MS can alleviate and/or avoid the endless build-test-break-fix cycles, common in many vehicle acquisition and testing programs of the past, thereby reducing costs and shortening milestone schedules. Although it’s difficult to quantify overall life cycle impacts resulting from doing it right the first time, MS has proven itself to be an excellent tool for decision makers, in response to limited time and budget constraints, as well as, aggressive procurement schedules. Although the current modeling methodologies are robust and being heavily utilized, there is still a lot of room for improvement of the modeling methodologies. The university-based research consortia sponsored by the Army are producing dramatic increases in modeling capabilities in areas such as driver models, engine research, and alternative energy sources. The Army is relying heavily on these tools in the future when it will be looking at advanced ground vehicle systems and their technologies.

REFERENCES


3. Supervisory Control of Hybrid Vehicles, H. Pen, University of Michigan, SPIE paper number 6228-12, 16 April 2006.


BIOGRAPHY

Mr. Michael Letherwood was born in Detroit, MI and obtained his engineering degrees from the University of Michigan and Wayne State University. He works for the US Army’s Tank Automotive Research, Development, and Engineering Center and is the Team Leader of the Automotive Research Team. Mr. Letherwood has been working in modeling and simulation of ground vehicle systems for over 15 years and is currently manages a large number of university research programs.
Simulation rules and ontologies as integral components of decision support systems

Tobias Lehmann, Robert Schmied, Andreas Karcher
Universität der Bundeswehr München
Werner Heisenberg Weg 39
85570 Neubiberg, Germany
E-mail: {tobias.lehmann/robert.schmied/andreas.karcher}@unibw.de

ABSTRACT

The following article describes how simulation rules can be described in an Web Ontology Language (OWL) ontology and be used for various manners in third party applications. Especially the analysis of rules connecting concepts within the Knowledgebase (KB) are of interest when planning support systems try to find a way to archive given objectives by combining formalized actions. In this particular case an OWL KB (enriched by Semantic Web Rule Language (SWRL) statements) becomes converted into a Prolog system after the reasoning process took place and projects inferred class structure into the rule based world. Finally an outlook is given how these transformed elements can be combined and enriched in order to work as a decision support system.

SIMULATION AND ONTOLOGY?

The central question of this paper is how useful simulation systems can be for ontology based decision support (and vice versa). Both are highly formalized but serve strongly different purposes – ranging from modelling, testing and forecasting behaviour to describing structures of real world aspects. One aspect could be to use ontologies as databases for simulation. This question raises performance questions immediately since most Models used (such as Jena) provide low performance only (Dickinson 2006). But the idea of modelling the subject of analysis by using Ontologies and therefore benefit from constantly growing libraries of all kinds of already modelled real world aspects seems promising (Miller, Swick et al. 2001). As depicted in Figure 1 ontology-systems aim to use upper standardized class definitions as a common base to enable the exchange of more detailed Midlevel or Domain Ontologies (Niles and Pease 2001).

INTRODUCTION – EBO CONTEXT

The Effects Based Operations (EBO) is the approach to consider not actions and objectives only but further more a concept named Effect. An effect can be described as the change between two situations within a focus area (Smith 2003). Effects arise from certain actions or situations and changes and they invoke further (different) effects building effect chains. These chains become recognized and documented in lessons learned during combat or peace supporting operations (Rickerman 2003). So the decision of future military leaders could be not objective orientated only but more focused on which effects might be desirable (Deptula 2001). Complexity of decision making grows over proportional because jointness and interagency relations become more and more important. This causes a shift from pure military operations to missions strongly influenced by and depending on diplomatic, economic and social affairs (Bingham 2001).

In order to support operational military leaders in their daily decision processes an information and rule integrating KB can bring light into the darkness of information flooding. But the wish for a machine making decisions in a highly volatile context filled with uncertain and missing information like the military environment is likely to be unfulfilled for quiet a long time due to knowledge acquisition problems (Baumeister 2004). Instead of this the human factor as the decision maker is to be supported with correct, relevant and timely well provided information. Therefore it is – in the context of decision support – not of the highest importance to develop a simulation tool which covers all aspects of a focus area but more to show relations between concepts and provide information from various data sources such as public search engines, secret service databases or mission reports.

Figure 1 Upper, Middle und Domain Ontologies
Source: see (Ševčenco 2004)

This interchangeability combined with flexible class structures based on Description Logic assertions turn ontologies into a very powerful tool to model data structures of real world aspects. But this upside becomes its greatest downside as well when it comes to connecting executable programs to the ontologies. Systems like F-Logic or JESS (Friedman-Hill 2003) provide options to connect functionality with data structures and therefore combine these two fields natively but have a lack in sense of exchangeability and most importantly in sense of description logics (Angele and Lausen 2003). Another aspect aims at the modelling of behavioural rules which are available within simulation systems in large quantities and strongly needed to model relations between concepts of KBs.
Analyzing Described Rules

Independent of the quality of knowledge it is possible to model truth-oriented and action-oriented rules. The truth-oriented approach defines static structure such as relations between concepts while the action-oriented approach is applicable for describing dynamic behaviour.

Truth-oriented rules are described in terms of SPO (subject-predicate-object)-triples. If one wants to define just concepts, predicates as “exists” are required – but no objects. It is furthermore sufficient to define “person exists” in order to establish the concept “person” (Dassow 2005). Therewith you can describe the whole static structure.

Action-oriented rules are described in terms of a precondition and an execution part (action expression). This is independent of the semantics of an action. In principle there are – as depicted in Figure 2 – three types of actions: own actions, third party actions and KB actions. The latter is comprehensible in terms of modelling laws of nature. Third party actions and KB actions are so-called reactions and are separated from own actions by class. In the same context the third party offers the possibility to create an alternative view. As a generic feature of this model one has to swap the own actions and the third party actions to get the counterpart model view which the third party uses.

In the context of decision support systems rules serve multiple purposes. They enable the description of dynamic behaviour and they formalize relations between concepts (Bryghta and Müller 2004). Depending on the underlying formal model of the simulation, rules can be accessible in a distributed application via network. In addition they can be identified using an URI or be part of an application database or script. If rules are identified by an URI (for example in the Web Ontology Language OWL or the Semantic Web Rule Language SWRL) then a unambiguous web wide exchangeability is established (Horrocks, Patel-Schneider et al. 2004). Therefore a wider spectrum of available and standardized models grows in existing publicly opened online libraries. The integration of rules into a KB itself turns out to be an additional benefit because traditional rule engines such as the Java System Expert Shell (JESS) have – during implementation time – no direct connection to the content of the KB a user works on (Friedman-Hill 2003). Keeping this upside in mind EBO rules have been modelled in RDF/OWL as common Event Condition Action (ECA) statements which can potentially be transformed into any systems language using XSLT or Jena Model.

In order to analyse the dependencies between concepts in the KB and use them to determine parameter and the KB including all its rules is transformed into a prolog program. This program provides additional methods to find out contradictory or similar objectives and quantifies effects to be taken. Furthermore general parameter, which potentially influence the objectives are determined.

Formalizing Actions

In order to combine actions and test, if a combination leads to a certain set of objectives these actions are to be formalized in a special way meeting the requirements of situation calculus described in (McCarthy and Hayes 1969). An action contains a precondition to check if it can transfer situation S1 into Situation S2. This precondition is a specialization of a rules condition. While a rule condition tries to match a situation (and then maybe fires) the action’s precondition contains information about its general applicability to a certain situation. This usually means that rules are more focused on instances while actions are focused on class level. As an analogy to the usage of reflection mechanisms in the OO world these two different modelling layers can be combined in OWL Full by setting classes as values of properties (McGuinness and Harmelen 2004). As an example an action keepUnderSurveillance makes surely sense on special infrastructure or landscapes only.

Benefit

Describing concepts and their relations in OWL is rather comfortable, since its flexibility in mixing structural and instance elements is unlimited, relations can be source or target of other relations (reification) and exchange and integration of existing models is efficiently supported by tools – although it is still a very hard semantic problem (Ehrig, Staab et al. 2005). In addition to this OWL can contain resolvable Description Logic statements which can lead to context dependent classification of concepts within the KB (Smith, Welty et al. 2004) and can therefore deal with some aspects criticized about OO (Jacobs 2005). The most important upside of an ontology representation can be considered the ability to infer similar structures (Ehrig, M., Y. Sere 2005). This enables a highly (but not completely) automated comparison and mapping of distributed and heterogeneous ontologies and use them as if they were created homogenously. Since simulation systems (such as SOAR (Lewis 2001)) usually formalize its components’
behaviour using rule like expression, these concepts and (especially) rules can be extracted and converted into OWL in order to aid the information support for military leaders.

IMPLEMENTATION EXAMPLE

The following part gives a brief insight into the usage of rules and actions as used in the information retrieval process of an effects pre-selection process. Exemplary subject to research are a set of objectives and three transport systems (see Figure 3). The objectives are to supply refugee camps with food, to keep the supply level of own troops (with ammunition, fuel and replacement parts) at a sufficient level (which hides exact quantities) and to minimize border crossing of drug transports.

<table>
<thead>
<tr>
<th>Objective</th>
<th>Transport system</th>
</tr>
</thead>
<tbody>
<tr>
<td>Supply refugee camps with food</td>
<td>Truck / railway, UN, NATO, Termiz - Maze-e-Sharf</td>
</tr>
<tr>
<td>Keep supply level of own troops above</td>
<td>Truck / helicopter, NATO, Termiz - Maze-e-Sharf</td>
</tr>
<tr>
<td>not sufficient</td>
<td></td>
</tr>
<tr>
<td>Minimize border crossing drug transports</td>
<td>Truck, car, pack animals, Wallodas / traficjob, Termiz</td>
</tr>
</tbody>
</table>

Figure 3 Objectives and Transportation systems

Each of these objectives aims directly at a transport system which can be described (for demonstration purpose with reduced attributes) by a transport medium, an organisation responsible for the transport and a set of routes determined by (here) one starting point – if known – and an end point. These concepts can be seen as a pure data structure, where the Description Logic offers means to classify a transport system as a system which has a cardinality restriction of at least one instance for each property. They can directly be imported from ontologies available online (e.g. http://swoogle.umbc.edu/). But still rules to combine these concepts at behavioural level are missing. Using rules from the German combat simulation system JOANA (Joint Operations Army Airforce) some relations are defined already:

- Effects of missing supply over time for civilians and own troops
- Trafficability of roads, bridges and other public infrastructure
- Effects of weapons on infrastructure and soft and hard targets
- General actions of military units (such as monitor, protect, control, march etc)

Next to other mechanisms firstly all parameter of the three objectives can be defined. As depicted in Figure 4 a triple based Prolog function parameter is defined as the union of the targeted instances of an objective and all instances which are mentioned in a condition part of a rule that effects at least one objectiveInstance or an other instance which is part of the condition of a rule which effects any objectiveInstance. In this context an objectiveInstance is an instance, which is targeted by an objective (e.g. the food supply). This recursion can be limited to a concrete number of levels in order to determine sufficient dependencies at acceptable costs in sense of computation time.

After this determination a decision maker has partly (surely not complete) knowledge about classes and potential instances able to effect the achievement of his given objectives. These instances or classes can be – emphasising and using experience and the combinatorial capability of a planner – dynamically linked to content of other KBs or search engines (e.g. by passing on name and class of an instance as get parameter of a URL search string) and therefore provide knowledge of different agencies or units. By resolving all objective and comparing their (defuzzified) quantities contradictory and depending objectives can become clear. This is of special importance since a military leader usually does not (and due to their numbers should not) know about all objectives within his area of responsibility.

Finally the delta between the current situation and the situation described by the objectives can (if numerically expressable) be calculated – the sum of all desired effects is now at hand. One can now try to calculate potential “solving paths” (in sense of combination of actions) using a Golog interpreter (including the previously mentioned formalized Actions). Performance problems due to a too complex context in sense of too many relevant individuals or too many steps necessary to archive all objectives set boundaries to this process.

RESUME AND FUTURE WORK

This process still lacks of many detailed steps which are partly under development and partly not subject of a current more conceptual project. But since military organisation within the NATO usually (except the US) have currently no sophisticated and fielded computer based decision support systems but simulation systems this formalization can be considered a first step. Things do be done within the close future are firstly the implementations of more interfaces between third party information systems and concepts within the KB and secondly to formalize the fuzzy and missing information aspect more intensely. Thirdly a rating for trustability of information in sense exclusion of wrong or not trustable information or of best practices is to be modelled.

REFERENCES


United States Army Command and General Staff College.


LARGE SCALE SIMULATORS
The Flight Simulation Environment of The University of Naples

Domenico P. Coiro (*), Agostino De Marco (*), Fabrizio Nicolosi (*)

(*) University of Naples “Federico II” – Dipartimento di Progettazione Aeronautica
Via Claudio 21, 80125 Napoli
Web: www.dpa.unina.it/adag/
E-mail: coiro@unina.it ~ agodemar@unina.it ~ fabrnico@unina.it

KEYWORDS

ABSTRACT
This paper presents the work done to set-up and test a real 6DOF flight simulation facility recently acquired. The whole system has been designed to be operated both as a driving simulator and as a flight simulator. The simulator is a full scale one, including real vehicle mock-up, a motion system, and a large projection system. Car half-body and aircraft cockpit are exchangeable and they can be easily installed on the motion platform.

The authors have worked mainly on the flight simulation side. The flight simulator cockpit has been conceived as a generic cabin of a small aircraft. The chosen software module that guides the various components of the system is based on FlightGear (Schgal et al. 2002), a civilian open-source flight simulator comparable to Flight Simulator from Microsoft.

The simulation of aircraft motion, the cockpit instrument panel and flight controls, the outside scenery are all managed by a number of instances of FlightGear talking to each other via net protocols. Moreover, the simulation is supported by two other software modules that control: (i) the motion platform, in conjunction with the external view generation module, in order to give a proper acceleration feel to the user, and (ii) a force reproduction system on the cockpit controls that adds realism to the piloting task.

Details about all these modules are given in the paper, along with the discussion of the advantages and potentialities given by the source code accessibility and the high configurability of FlightGear. Particular effort has been spent to implement hinge moment equations in the simulation software in order to be able to reproduce a very realistic pilot effort and to obtain a reliable closed-loop force-feedback system on all aircraft commands (elevator, ailerons and rudder). The system is also able to reproduce force-free elevator or stabilator response.

INTRODUCTION
The authors have worked to the specifications, the development and the final acceptance procedure of a recently acquired simulation facility. The whole system has been designed to be operated both as a driving simulator and as a flight simulator and is going to be managed by two different research teams, one including the authors, the other coming from the Transportation Department of the same University.

The system is a full scale simulator and includes real vehicle mock-up, a motion system, a large projection system, and force feedback modules. The half-body of a real car and the aircraft cockpit mock-up are exchangeable and easily installed on the motion platform. Apart from those characteristics shared with the automotive simulator team, the authors have worked mainly on the flight simulation side.

SIMULATOR LAYOUT AND COMPONENTS
The simulator is installed in Naples in one of the buildings belonging to Consiglio Nazionale delle Ricerche (National Research Council, CNR), Istituto Motori. The simulator room layout is shown in fig. 1.

![Figure 1: Simulator room layout](image-url)
The flight simulator cockpit has been conceived as a generic cabin of a small aircraft because the principal aim of this facility is to make investigations on flying qualities of light and ultra-light aircraft and to offer a training options to the pilots of such airplanes.

As seen from the figure, the building is divided into three areas: the simulator main room, the supervisor room, the briefing room. The top-view shows the horizontal motion envelope of the cockpit, the main room dimensions, and other minor details. Also shown are the three large screens located in front of the cabin where three DLP projectors (DS30 from Christie Digital; 3000 lumens, 1280x1024, SXVGA) project a composite image of the virtual outside environment. This particular projection system, fig. 2, is preferred in car simulators and proved to be effective in the flight simulator presented here. The reproduced image corresponds to a 190° field of view (horizontal).

**Figure 2: Projected images in front of the moving cabin during simulation**

The subject pilot positioned in the cabin is given a motion cue during the simulation. The cue is obtained by animating the airplane mock-up with a six-degree-of-freedom motion platform, fig. 3.

**Figure 3: Motion system adaptor plate mounted on the top of the motion platform**

Since the simulator can be used for driving simulation or flight simulation, the two available cockpits can be plugged on the motion platform. A primary adaptor plate is fixed to the motion system. Each cockpit is mounted on a secondary adaptor plate, which is designed to be firmly coupled with the primary one. The primary plate is shown in fig. 3, on the top of the motion platform.

The car and aircraft cockpits are switched using a fork-lift truck. The entire operation takes less than 1 hour. The used cockpit is safely plugged on the motion system adaptor plate shown in the same figure. The mock-up of the aircraft cockpit is shown in fig. 4.

**MODE OF OPERATION**

The effective use of the simulator requires the presence of a person in the supervisor room, who is in charge and responsible of ensuring the safe use of the system. An emergency stop button in this room enables the supervisor to safely stop the simulation in case of emergency. The simulator room access doors are equipped with electrical contacts to ensure that they are closed and a red/green light is set outside the simulator room to prevent anyone to enter the room when the simulator is running.

**Figure 4: Mock-up of the aircraft cockpit mounted onto the motion platform**

The supervisor has a direct sight on the simulator area, as seen from fig. 5, and can stop the simulation if the security rules are not respected.

**Figure 5: View from the supervisor room**

The pilot inside the cabin is protected by electrical contacts on the cockpits doors, that have to be closed during the simulation, by a safety belt that has to be locked during the simulation, and by an emergency stop button onto the dashboard. A camera (audio/video) is installed inside the cockpit to have a sight on the subject and ear the sound within the cockpit from the supervisor room.
The simulator motion area is protected to prevent anyone or any object to interfere with the simulator. The motion area is painted on the ground. The access stairs are linked to electrical contacts when they are at their parking places. These contacts are taken into account by the system. The simulator will be certified in conformity with the national and European community rules.

GENERAL ARCHITECTURE
The flight simulator cockpit reproduces a generic cabin of a small aircraft, fig. 4. The cockpit structure consists in an aluminum skeleton in which synthetic material (such as polycarbonate) sheets are slipped into the bones of the skeleton by means of appropriately designed slides. This solution has proved to be more convenient than having a real aircraft cockpit, both in terms of cost savings and of simulator usability.

Inside the cockpit, the main instrument panel consists in two tactile LCD screens. One screen is used to display a virtual flight panel. The second screen enables the display of what is needed by the experiments, such as moving maps or flight parameter real-time plots. A space is also reserved for a third screen. A picture of the dashboard with the two screens in use is shown in fig. 6.

Figure 6: Virtual indicators and moving map inside the aircraft cockpit
The flight controls consist in: a Cirrus II Flight Console from Precision Flight Inc., a yoke, which is included in the original flight console but whose position has been conveniently modified, and a pair of rudder pedals. A picture of the controls is shown in fig. 7.

Figure 7: Flight controls
To simulate the ATC radio link, the subject pilot has a headset with a microphone.

The yoke and the rudder pedals are equipped with a force feedback system giving to the subject an additional cue on the piloting effort. Fig. 8 shows a detail (control panel back view) of the mechanical linkage connecting the yoke to one of the actuators of the force feedback system.

Figure 8: Detail of yoke loading leverages
COMPUTER AND SOFTWARE ARCHITECTURE
The overall simulator computer architecture is represented in fig. 14. The scheme reports the main flows of signals from the supervisor room computers to the rest of the simulator hardware. The layout is optimized with respect to the performance required by each piece of software used by the system.

The design of the wiring, needed for hardware interconnections, has been optimized with respect to the ease of use of both car and aircraft. The external wires of the cockpits are plugged on a dedicated interface on the primary adaptor plate, fig. 9.

Figure 9: Interface between external wiring and internal hardware
When the cockpits are swapped, the wires are unplugged from the formerly utilized cockpit and then plugged to the new one in use.

The chosen software module that guides the various components of the system is based on FlightGear, a civilian open-source flight simulator comparable to Flight Simulator from Microsoft.

The software is decomposed into various modules:
- a dynamic model of the airplane; the one currently chosen is JSBSim (Berndt 2004);
- PLIB, a portable API that contains the base functions of the graphical and sound environment;
- SimGear, the main management module which sits on top of the PLIB library and controls the simulation and the scene construction;
- TerraGear, a set of tools dedicated to the terrain generation, importing GIS data, DTM, etc.
The image generation in FlightGear allows displaying both 3D and 2D. The main functions on top of classic 3D image generation software are:
- Displays of the moon, stars and sun.
- Head Up Display.
- Interactive dashboard display.
- Clouds, wind

The images are generated using PC technology, by the graphic board NVIDIA GeForce Quadro FX4000. The simulation of aircraft motion, the cockpit instrument panel and flight controls, and the outside scenery are all managed by a number of instances of FlightGear talking to each other via net protocols. Moreover, the simulation is supported by two other software modules that control: (i) the motion platform, in conjunction with the external view generation module, in order to give a proper acceleration feel to the user, and (ii) a force reproduction system on the cockpit controls that adds realism to the pilot’s task.

**MOTION SYSTEM CHARACTERISTICS**

The motion system is based on a six-degree-of-freedom motion base Maxisim 610-450-16-12 from cueSim, fig. 3 and fig. 9. The motion base, or motion platform, consists of a top frame, bottom frame and six high efficiency electric actuators arranged in the Stewart platform format. The payload is mounted onto the top frame.

The actuators are secured to the frames by precision manufactured pivots to provide a stiff, stable platform. Each actuator comprises a precision ball-screw directly driven by a brushless servomotor with position feedback, fig. 10.

![Figure 10: Motion base servomotor detail](image)

The combination of sub-micron position resolution, high actuator peak thrust and low backlash allow high-fidelity motion with low tracking errors. The absence of belts or gearboxes in the drive system produces smooth, quiet motion. The actuators are oil lubricated and require the minimum periodic maintenance.

The main characteristics of the motion base are the following:
- Max. payload: 1000 kg
- Height (min/max): 1180/1675 mm
- Mass: 625 kg
- Surge position: -491/+432 mm
- Surge velocity: 718 mm/s
- Surge acceleration: +/- 1.39 g
- Sway position: -425/+425 mm
- Sway velocity: 712 mm/s
- Sway acceleration: +/- 1.2 g

- Heave position: -247/+248 mm
- Heave velocity: 484 mm/s
- Heave acceleration: +/- 0.59 g
- Roll position: -25/+25 deg
- Roll velocity: 50 deg/s
- Roll acceleration: 575 deg/s/s
- Pitch position: -24/+24 deg
- Pitch velocity: 48 deg/s
- Pitch acceleration: 595 deg/s/s
- Yaw position: -43/+43 deg
- Yaw velocity: 82 deg/s
- Yaw acceleration: 1100 deg/s/s

**CONTROL FORCE FEEDBACK**

The flight simulator is provided with a dedicated force feedback module, also called “control loading” module, that reproduces the piloting efforts on the yoke (push/pull, turn left/right) and on pedals, according to the simulated flight condition.

Simulation of a pilot-in-the-loop general manoeuvred flight requires the solution of the classical set of airplane equations of motions, coupled with an auxiliary set of equations for the attitude and flight path determination. This is done by FlightGear, in particular by the chosen aircraft dynamic model, which in our case is JSBSim.

When a force-feedback system is matched with the aircraft equation solver, the cockpit control loads have to be computed from the known aircraft state at each simulated time step. Stick and pedal loads are controlled with a given frequency, typically higher than the simulation frequency, and properly reproduced (see Rolfe 1988 for a general discussion).

In our control loading module the simulation of the aerodynamic control surfaces (aileron, elevator, rudder) has been extended, with respect to FlightGear’s functionalities, and implemented in a dedicated piece of software (ForceGear in fig. 15). The evaluation of the aerodynamic and inertial actions on the aerodynamic control surfaces is one of the main tasks of this code. Their equations of motion are solved within the control algorithm loop between two successive FlightGear time steps. The aerodynamic/inertial actions are evaluated in what is also referred to as the “inner” integration loop. FlightGear’s job is then called the “outer” integration loop.

In the general case of simulated stick-free manoeuvred flight the additional unknowns are the time histories of surface angular excitations: $\delta(t)$ (right aileron), $\delta(t)$ (elevator or stabilator), $\delta(t)$ (rudder). The “stick-free” conditions are those particular situations in which the pilot actions on the cockpit controls are null and the aerodynamic control surfaces are free to float under the effect of external actions. These actions depend: (i) on the aircraft motion and acceleration, and (ii) on the characteristics of the mechanical linkage between the control column located in the cockpit and the tail plane moving parts. In all cases the excursions time rates are evaluated and used by the force feedback module.
Elevator Dynamics

- Inertial coupling actions
- Typical inertial term
- Aerodynamic hinge moment
- Pilot’s action

Figure 11: Example of model equation of elevator control dynamics

An example of how the generic longitudinal control dynamics is modelled is shown in fig. 11 (e stands for elevator). The first typical inertial term is the moving surface angular acceleration multiplied by the surface moment of inertia $I_e$. The remaining terms are all torques acting on the elevator around the hinge (hinge moments, $H$). The term $H_{e,\text{In}}$ represents the moment resulting from the inertial coupling actions (time-varying aircraft pitch rate and/or combination of non-zero roll and yaw rates about airplane center of gravity). The term $H_{e,A}$ is the moment resulting from aerodynamic (A) actions on the elevator. Finally, $F_{e,C}$ is the force applied by the pilot (C stands for commanded) to the yoke. It is reduced to a moment about the elevator hinge after dividing by the dimensional gearing ratio $G_e$. In stick-free condition this term is zero. Generally, in a commanded manoeuvre the pilot control force $F_{e,C}$ is non-zero and is treated as an input in the model of fig 10. The algorithm controlling the force due to the pilot measures the action actually exerted on the yoke, evaluates the inertial coupling and aerodynamic terms, and reproduces the angular acceleration contained in the first term.

Figure 12: Schematic of longitudinal control loading

At the beginning of a generic control loop, the forces and torques actually exerted on the cockpit controls by the pilot are measured by dedicated load cells. The current yoke and pedal positions are sensed by a number of corresponding potentiometers and optical encoders. By evaluating the difference between the sensed actions and the calculated aerodynamic/inertial actions, the control loading module orders the yoke and pedals to move (accelerate) accordingly, giving the desired “feel” to the pilot. If pilot’s action is adequate to react to the feedback and keep the yoke/ pedal position stationary, the flight conditions remain “stick-fixed”, or nearly so. If not, the unbalance between the force actually exerted on the control and the one calculated by the force-feedback system from simulated flight data results in a general manoeuvred flight with a varying excusion of one or possibly all the aerodynamic control surfaces. The actual amounts of the excursions, in terms of yoke and pedal displacements, are fed back to FlightGear and used in the successive outer integration step.

A schematic representation of the control loading process for the longitudinal command (yoke push/pull) is reported in fig. 12. The load, denoted in figure as $G_e(H_{e,A}+F_{e,C})$, is the stick-reduced force resulting from the aerodynamic/inertial-coupling effects on the horizontal tail plane moving part. They are evaluated and compared with the measured pilot effort.

Two useful generalizations have been implemented in the inner loop model. The first includes the effect of the mechanical linkage dynamics on the control surface motion. The user is able to specify the equivalent reduced masses that model the motion and inertia of the actual command line of the simulated aircraft. The second generalization includes the effects on the control displacement due to the mechanical friction and to the presence of springs. The user can specify an appropriate friction damping coefficients and the stiffness of a springs possibly located along the command line.

The geometric, mass, inertia characteristics of each control surface and the hinge moment coefficients are managed by the control loading software.

At each integration time step the needed coefficients, for instance the elevator hinge moment coefficient, are retrieved from a suitable database with a table look-up procedure. The entries are a number of given parameters of the motion, such as the aircraft angle of attack and airspeed. The latter are taken from one of the data structures exposed by the main simulator code via a socket protocol. This is one of the most important capabilities of FlightGear: to allow the interfacing with an external client code like our control loading software running onto a different dedicated computer.

The computer program driving the force feedback system is configured using input files in XML format. The configuration style follows the FlightGear and JSBSim philosophy. This choice guarantees a level of uniformity with the main simulator configuration format and allows a smooth transition to possible future integrations with newer releases of FlightGear. The input file format is well structured in its nature and allows an easy and clear definition of aircraft control systems and their operating characteristics, according to the particular control model implemented.

The configuration contains a description of the following items: control surface geometric and mass properties (shapes, dimensions, hinge position and orientation, eccentricities, moments of inertia), control system...
mechanical properties (gearings, friction factor, spring stiffness), control surface aerodynamic characteristics (hinge moment aerodynamic coefficients), control surface auxiliary characteristics, data logging parameters.

The actuators and the rest of the hardware of the force feedback system have been chosen in order to reproduce a realistic amount of effort required to the subject pilot. The following are the main characteristics:

- Max. force on yoke $\pm$ 400 N (push/pull)
- Max. torque on yoke $\pm$ 40 Nm (turn left/right)
- Max. force on each pedal 400 N

When needed, the force feedback can be easily disconnected.

Fig. 13 shows an example of simulated manoeuvre for a Cessna C172 aircraft. Both longitudinal and lateral-directional motions are induced, as seen from the variation of elevator and rudder angular positions. The efforts plotted in figure are the forces applied on the yoke (left) and pedals (right) by the pilot. Their magnitudes are within the range of predicted control force in steady flight at the same mean speed.

CURRENT FLIGHT SIMULATION WORK

Fig. 15 depicts the interconnection between flight tests and simulation research topics investigated by the authors and the simulator operating characteristics. As shown in the top-left side of the figure, to carry out a flight simulation of a given aircraft one has to collect a number of data coming from: (i) flight tests, (ii) wind tunnel experiments, (iii) numerical or semi-empirical estimations. Experiences gained by the authors in these fields are reported in (Coiro et al. 1998, Giordano et al. 2001, Coiro, Nicolosi, De Marco 2002, Coiro, Nicolosi, De Marco, Genito 2002, Coiro 2003, Iscold 2004). In particular, the authors have worked at the certification process of the G97 ultra-light aircraft (fig. 15, top-left, see Giordano et al. 2001). All aspects regarding JAR-VLA certification procedures have been object of research. Accurate and detailed analysis of flight test manoeuvres have been performed and comparisons with numerical predictions have been done. All G97 performances have been measured. Particular attention has also been given to the parameters estimation for the complete aircraft aerodynamic and dynamic characterization.

As seen from the fig. 15 (top center) these data are properly structured in XML format according to the FlightGear configuration style.

The work dedicated to the force feedback model is highlighted in the bottom/right part of the figure. The geometric, mass, inertia, and aerodynamic characteristics of the control surfaces along with the mechanical properties of the control transmission are taken into account in the force cue to the cockpit controls via the force feedback module. This is driven by the ForceGear software designed by the authors and implemented in cooperation with Oktal, the French company who set up the whole system.

CONCLUSIONS

The work presented introduces the simulation facility of the University of Naples. For the first time it is presented a simulator using the open source flight simulation software FlightGear in conjunction with both a 6DOF motion base and a force feedback module based on the aerodynamic control surface simulated motion.

REFERENCES


JSBSim website, http://www.jsbsim.org/

Figure 14: Simulator computer architecture
Figure 15: Flight simulation research topics investigated by the authors and their connection to the simulator characteristics
LABORATORY TOOLS FOR USE IN SIMULATION
BASED DEVELOPMENT OF ADVANCED MILITARY RUN FLAT TIRE SYSTEMS

Aleksander Kurec, Jose R. Mabesa, Jr.
U.S. Army Research, Development, and Engineering Command
Tank-Automotive Research, Development, and Engineering Center
(RDECOM-TARDEC)
AMSRD-TAR-N/157
Warren MI 48397-5000
Tel: 586-574-7583 Fax: 586-574-8667
e-mail: aleksander.kurec@us.army.mil

Abstract

Recent advances in bulk material and material processing has lent itself to broader applications. The U.S. Army positions itself to leverage this technological advancement, and reap its benefits. Using the RDECOM-TARDEC’s Ground Vehicle Simulation Laboratory (GVSL), research scientists and engineers are using these both physical and analytical tools to develop advanced military grade run flat tires for their vehicle platforms. Basic research development is conducted to adequately evaluate and optimize every design variable. To date, the U.S. Army is characterizing, modeling, simulating, and optimizing their prototype fielded design for the High Mobility Multipurpose Wheeled Vehicle (HMMWV) platform. In this paper, we will discuss the RDECOM-TARDEC GVSL’s use of their advanced laboratory tools to develop an advanced military-grade run flat tire system. Utilizing partnerships with other government laboratories, academia, and private industry, the US Army research scientists and engineers will be able to provide the best run flat solution for their customer, the US Army Soldier.

Introduction

The need for an Advanced Military Run Flat technology exists in current commercial and military vehicle marketplaces. Run flats systems have been used for a number of decades worldwide. Both markets have needs that address safety, mobility and load capacity requirements. The maturity of this technology has identified critical area’s, such as, performance, reliability, serviceability and a level of acceptance. Each market has established detailed preferences after a history of use and longevity. Basic pneumatic tire Run Flat technology can easily be separated into two major groups (1) innovative tire construction and (2) tire cavity inserts. Both groups have market acceptance.

In early commercial applications the main use only addressed safely stopping the vehicle and having an ability to travel a short distance at a relatively low speed. As demands increased so did the need for greater distances at higher speed, such as those encountered in military use: a soldier in harms way wants to keep moving as fast as possible thru various terrains for as far as needed. This type of demand requires better materials for higher dynamic loads, a configuration that would withstand greater stresses caused by driving while flat.

The present Run Flat systems are not adequately reliable for either tires or inserts. The introduction of innovative sidewall tire technology has begun to be accepted. However, rough rides are inherent with stiffer sidewall tires and drivers recognize that tire characteristics have changed. The predominant change has been in steering response. The stiffer sidewalls are constructed with a semi-rigid plastic disc that limits tire deflection. After loss of pressure, the semi-rigid sidewall reduces tire deflection and provides partial support preventing the tire from fully collapsing. However, there are two major differences between the technologies, inserts are wheel dependent and run flat tires are sidewall construction dependent. Inserts mounted in the tire cavity also limit tire deflection during or after the decreased pressurization. Tire cavity inserts could be rated higher on reliability than tire, since the inserts provide support and mobility after the tire has failed. Run flat inserts are mounted inside the tire cavity allowing possible reuse and the use of standard tires. Run flat inserts are designed to support higher radial and lateral loads, which allow greater versatility for vehicles and equipment.

Presently, the U.S. Army approved and utilizes a “Donut Style” tire cavity insert for its Run Flat system on HMMWV vehicles. The current insert has been tested to specific load ratings and operated for a set distance while deflated. During testing, the tire first will begin to bundle under the tread, generate high stresses on the shoulder (corner) area and destruct within a short (approx. 5 mile) distance. The U.S. Army test specification ATPD 2099 C [2] requirement states a 30-mile test on three different terrain types:

<table>
<thead>
<tr>
<th>Distance (miles)</th>
<th>Terrain Type</th>
<th>Average Speed (mph)</th>
</tr>
</thead>
<tbody>
<tr>
<td>12.0</td>
<td>Cross Country</td>
<td>12.0</td>
</tr>
<tr>
<td>9.0</td>
<td>Secondary Road</td>
<td>21.0</td>
</tr>
<tr>
<td>3.0</td>
<td>Paved Road</td>
<td>30.0</td>
</tr>
</tbody>
</table>

The ability to maintain mobility after completing the 30-mile test meets the specification.

The final acceptance includes evaluating in-field replacement or repair of the Run Flat tire/wheel assembly. After removing the wheel assembly from the vehicle, the tire is replaced. Presently, the disassembly of the wheel requires special tools and a procedure that involves considerable effort and time to remove the “Donut Style” insert. Once removed, the re-installation of the “Donut
Style” insert exceeds more than 2 hours, per TACOM, they have identified this task to be an area that needs improvement.

It is apparent; the need exists for an upgraded compatible Run Flat System that meet the requirements for new applications.

Technology Advancement

Drive Dynamics, in partnership with the U.S. Army, has patented technology advances the science of Run Flat Insert systems by addressing factors involved in tire performance during inflated, partially inflated and flat running conditions. The major advantages of this system are,

- Positive attachment to the wheel for better support and torque transfer.
- Ability to withstand higher loads.
- The inserts provide:
  - Wider bearing surface.
  - Larger rolling radius.
  - Tread aligned support.
  - Crowned surface for flat tire centering.
- Easily installed by use of multi-segments.
- Inflated tire performance does not change.
- Tire integrity preserved for short distance run flat operation.
- Compatible with standard tires.
- Fully functional after tire has been shredded.

The design, development and testing verifies that the major advancements out perform current run flat insert systems. To accomplish the development, three design topics were identified as the basis for improving the system. These topics include tire sidewall deflection, centering of the tire to wheel load and determination of the Run Flat load.

Tire Sidewall Deflection

Inflated tires have a measurable loaded rolling radius which most vehicle designers used for driveline related calculations and performance specifications. These values can easily vary from loaded front and rear axle supported weight, which influence gear ratio values. On most all-wheel drive vehicles consideration of axle weight/gear ratio differences are corrected by changing or having different tire sizes or inflation pressures front/rear.

Changes in pneumatic tire spring rate are directly related to changes in inflation pressures. The operational limits of tire sidewall deflection can be monitored for best tire performance within the design weight and inflation values. Upon changes in tire cavity volume from adding inserts, the spring rate and dampening values also change.

The first “Key” to Run Flat insert systems would be determining an acceptable amount of tire sidewall deflection beyond the loaded rolling radius and the lowest inflation values. This establishes an approximate Outer Diameter (O.D.) for the insert configuration.

Centering the Tire-to-Wheel load

Pneumatic pressure within the tire cavity applies a uniform load to the rim portion of the wheel. Flange-to-flange distances establish a centerline of radial force applied to the rim while the location of the disk portion of the wheel establishes the moment to the axle or hub. Lateral loads are considered during steering evaluations and verified by actual test data.

The second “Key” to Run Flat insert systems would be designing a support device to provide a load reaction either uniformly or closely loaded to the center of the rim.

Determination of the Run Flat load

Both the tire and wheel have been designed to satisfactorily perform at specific load and speed levels and rated per those values. The vehicle designers select tires and wheels to meet their needs and generally use values that allow a 1.5 times or more safety factor to the tire and wheel whichever is greater. Each application varies and relies on an approval process that is verified by actual testing.

The third “Key” to Run Flat insert systems would be establishing a specific load range suitable to a selected tire and wheel assembly.

II. Drive Dynamics Advancements

Multi Segment Insert System

A Multi-Segment Insert System within the tire cavity offers the best configuration for installation and removal. This clone permits a greater number of design shapes to help support higher loads and center a deflated tire. Each segment can be engineered to withstand sharp impacts plus provide the required radial or lateral support. A lightweight polymeric material provides the resilient or elastic properties needed to meet higher load levels and cushion any transient stresses.

Positive Locking Torque Ring

A Positive Locking Torque Ring mechanically fastens the inserts to the wheel creating a structural path. The benefit of “Positive Locking” allows for better traction and braking during a run flat condition. The resulting non-slip action assists in controlling and handling the vehicle while steering. Fastening the system together better maintains balance and concentricity during long periods of inflated use. Fatigue evaluation [6] of steel would make it the preferred material for use. By design, all loose parts are done away with that normally induce vibration at higher speeds.

Non-Bead Lock Approach
Advancement in fastening allows the designer to use either a “Non-Bead Lock Approach” or conventional bead locks. Preferred non-bead locking the tire sidewall allows the deflated tire to be re-inflated more often without tread shoulder damage as does conventional bead lock systems. Safer operation occurs when the tire slips on the wheel during Run Flat conditions and the insert rolls on the tire.

**Load Centering and Surface Crowning**

Drive Dynamics’ technology best replicates the tire-to-wheel loading by “centering” the inserts between the rim flanges. Knowing the steering system “offset radius”, distance being the camber pivot axis and mid point of the tire footprint, the centered and slightly crowned insert surface helps to maintain operation similar to that of inflated conditions. Any change in “offset radius” may cause a one-side pulling action and vehicle steering problem to occur.

**Ballistic Resistance**

Applications for engineered polymers have been used for decades in the tire industry, for example forklift tires, have demonstrated excellent durability and impact resistance. These same properties offer excellent ballistic resistance to both surface damage and actual bullet penetration. Drive Dynamics and BASF™ proprietary formulated polymers withstood ballistic impacts of a .308 caliber 50 grain, ballistic tip bullet. On the sidewall section, the bullet would enter and exit without tearing material or creating a large exit hole. The elastic properties allow the material to be deflected without any vertical shear as a bullet penetrated and exited.

**Assembly**

A major advancement in the technology focused on the ease of assembly. Most Run Flat Insert systems require special tools and a cautious procedure to follow. However, using a multi segment insert system components can easily be installed by hand with minimum instructions. Only small hand tools are needed for fastening and positive locking the system. The figures below demonstrate how quickly the system can be assembled.

Figure 1: Run flat components with the wheel.

Figure 2: Run Flat System placed over the wheel.

Figure 3: Insert segment placed in the tire.

Figure 4: Torque ring placed in the tire.

Figure 5: Fastening the inserts to the torque ring.
inspect or to make the decision to replace damaged inserts can be completed in a relatively short time frame. As tires are replaced the need to replace the insert should be evaluated by an individual trained in the technology.

**Test Direction and Protocol**

The Federal Motor Vehicle Safety Standards (FMVSS) [7] establishes specific test protocols that are accepted for approval of both on-road and off-road use. The U.S. Army has adopted FMVSS –120, “Tire Selection and Rims for Motor Vehicles Other Than Passenger”, for dynamometer testing as the test procedure for testing inflated tires with installed Run Flat Insert Systems. In addition, the U.S. Army specifies tests within system specifications for certain vehicles, for the HMMWV the System Specification ATPD 2099C represents the HMMWV A2.

The Drive Dynamics Run Flat Insert System was tested at U.S. Army RDEC-WARDEC in Warren, MI on two occasions, November 2003 and April 2004. The following test protocol (basis of modified FMVSS-120) was used for conceptual Run Flat Insert testing.

**Test Protocol**

Items to be tested:

- Tire/Wheel Assembly w/Drive Dynamics Run Flat Insert
- Tire Size: 37 x 12.5R16.5 LT Goodyear Wrangler
- Wheel Size: 16.5 x 8.25 2 Piece - 12 Clamp Bolts w/8 bolt circle
- Insert: 3 Piece Mounted to 16.12” dia. Single torque ring

**Test procedure:**

1. Install tire/wheel assembly on RDEC-WARDEC wheel dynamometer.
2. Set cold inflated pressure to 50 PSI.
3. Record temperatures @ start ambient, each 10 min/run.
4. Measure tire temperature ...Tread Crown, Sidewall, Rim.
5. Apply a 4000 lbs load and run @ 42 MPH for 30 minutes.
6. Apply a 4000 lbs load and run @ 48 MPH for 30 minutes.
7. De-inflate via removing valve stem core.
8. Apply a 4000 lbs load and run @ 30 MPH for 1 hour.
9. Upon evaluation of performance increase load to 4500 lbs for final test.

**Test Recording:**

1. Read and record all data on the above test procedure.
2. Note any problems, such as tire separations, excessive noise.
3. Take pictures as required to help identify problem areas.
4. Record test results.

![Image](image_url)  
Figure 9: Tested Run Flat System after radial loading and running at 4500 lbs.

**Discussion**

The U.S. Army recognizes the need to evaluate current equipment and systems that are used today for completing its higher demand missions. As part of the evaluation process, will the current available equipment meet those needs and be best suited for the soldiers? To maximize this process, the U.S. Army has an interest in technological advancements for mature applications and/or problematic systems. Higher demands from increased payloads on mobile equipment and need to be readily deployable have been under constant review.

Working together, the evaluation of Drive Dynamics new innovative approach to advance Run Flat Insert Technology has shown remarkable results from testing at TACOM-TARDEC in Warren MI. The system would achieve higher load capacities, out perform existing systems and assure positive locking. For the user, the new system would be better for installation, service and possible reuse. The U.S. Army expressed an interest in the technology and had entered into a CRADA with Drive Dynamics on the continuation of development. Dynamometer testing for both inflated and deflated (Run Flat) conditions has demonstrated that the innovative system would meet the projected needs of the U.S. Army. These features being:

- Operating loads up to 4500 pounds
- Larger rolling diameter 25 inches
- Rim loading Centered between flanges
- Mounting system Positive attachment to wheel
- Insert width Matched to tire tread
- Serviceability Field or depot, hand tools only

**Installation time**
- 15 minutes insert only
- 30 minutes complete tire assembly

These features are key advancements within the Field of Run Flat Technology.

**Conclusion**

This paper describes the innovation and development of an advanced Run Flat Insert System for Pneumatic Wheeled Vehicles. The advancement addresses military applications that currently use Run Flat Systems, such as the HMMWV. To increased safety, capacity and operational performance, a multi-segment design that positively locks to the wheel provides a number of benefits over current systems.

- Higher operating loads.
- Larger Run Flat rolling diameters.
- Flange centered rim loading.
- Positive-locking system for attachment.
- Easily serviced for in-field use or at depot levels.
- Significant reduced installation time.

Jointly, in development, the U.S. Army has entered into a CRADA with Drive Dynamics LLC of Dallas, TX to verify and validate the concept as a practical and ready to use system. The ongoing testing shows the system to be suitable on forward-looking applications.

**References**

1. The U.S. Army test specification ATPD 2099 C.
   Titled: Run-Flat Support for Pneumatic Tired Wheel.
   Titled: Run-Flat Support for Pneumatic Tired Wheel.
4. Pending Patent #60/559,856 Dated April 6, 2004  
   Titled: Run-Flat Support and Torque Ring Anchoring apparatus for Pneumatic Tired Wheel.
TRAINING SIMULATORS AND DISTANCE LEARNING
CONSTRUCTION MACHINERY DYNAMIC MODELS FOR REAL-TIME TRAINING SIMULATORS

Aron Pujana-Arrese  
Ana I. Martínez-Esnola  
Félix Martínez  
Carlos F. Nicolás  
Joseba Landaluze  
IKERLAN Research Centre  
E-20500 Arrasate/Mondragon, Spain  
E-mail: apujana@ikerlan.es

KEYWORDS  
Multibody modelling, multi-domain modelling, real-time simulation, training simulator, construction machinery.

ABSTRACT  
This paper deals with the approach used for the development of real-time dynamic models of four different types of construction machinery in the framework of the European Collective Research project “VAR-TRAINER”. The project objective is to create a training platform for construction operators focusing on aspects of safety at work and based on the use of real-time simulators. The use of commercially available tools for modelling and automatic code generation is addressed and the problems encountered are also discussed. The main components of the machine dynamic models are also described in the paper, and the solutions for realism and code efficiency requirements are explained.

INTRODUCTION  
The use of simulators as training tools for machine operators is spreading rapidly. Although some years ago simulators were mainly used for leisure and in the defence and aeronautics sectors, in recent years their use has also extended to new areas such as construction (Lorenzo et al. 2001), medical applications, the automotive sector, etc.

When the simulation of machines is related to people training, the realism of the simulation becomes a relevant aspect. This realism is closely related to the way in which the dynamic behaviour of the machine is modelled in reaction to the operator commands and to the interaction between the machine and the working environment.

It is also important to note a new trend towards the use of automatic code generation tools, to ease the process between the design and modelling phases and the real implementation and model execution in simulation.

This paper presents the modelling approach used by the authors for the development of dynamic models of complex construction machines within the framework of a construction operator training simulator environment. The technical aspects discussed and the modelling methodology described are also valid when dealing with machine dynamics modelling for other simulation applications such as HIL (Hardware in the loop) applications for early validation and verification of control strategies and control equipment (Landaluze et al. 2004).

Firstly, the paper describes the European “VAR-TRAINER” project in which IKERLAN has developed the dynamic models of four different types of construction machinery (wheel excavator, dumper, work goods lift and mast climbing platforms) underlying the performance requirements imposed on these models. Secondly, a description is given of the main technical characteristics of these machines, and the development methodology followed for building real-time simulation oriented dynamic models is explained. A detailed description of the dynamic model development process in Dymola for the machines within the project framework is then presented. Finally, there is a summary of the conclusions obtained from the work carried out.

DESCRIPTION OF THE VAR-TRAINER PROJECT  
The general objective of the project is to develop a versatile real-time simulator, based on the integration of mechanics, electronics, automated technologies, computer science and augmented reality in order to train workers in the use and operation of construction machinery. Special attention is paid to safety aspects with the aim of reducing the accident rate in the construction sector. In order to achieve this objective the leading edge technology in machine modelling for real-time simulation and mixed reality (virtual & augmented reality) has to be applied.

Although the use of machine simulators for training purposes is expanding nowadays, no commercial simulator exists for training workers in the use of construction machinery. Likewise, there is no certified training in matters of safety when using this machinery.

Most of the available simulators are specifically oriented towards a single machine or process type and are not open platforms that could be re-configurable to cover different
machine types, different levels of realism or different pedagogical aspects. Besides, machine training simulators usually aim to cover training in all the machine’s functionalities and are not focused on specific training points, such as safety issues, for instance.

As a result of the project, a simulation and training environment will be developed. This simulator will be sufficiently versatile and user-friendly to train in different types of construction machinery:
- Lifting equipment: work goods lift.
- Aerial work platforms: mast climbing platforms.
- Heavy works hydraulic machinery: wheeled excavator and dumper.

All of them will be possible with just one single simulator, thus ensuring its profitability and flexibility. This training program will be made up of three different training levels:
- A computer-based training (CBT) program, which will train the user in the rules to be observed in the interaction with and operation of construction machinery.
- A light simulator version which can be run on a computer (PC-based version).
- An integrated platform simulator version comprising the most advanced versions of the different hardware (machine cabin, operator interface and visual devices) and software modules.

The aim is to ensure the widest possible dissemination and impact by providing different alternatives (in features and cost) to the potential users (construction sector Industrial Association Groups and SMEs).

The training and simulation environment will focus on safety aspects in order to:
- Provide a safe training tool for workers and reduce the training time on real machines.
- Take a first step towards a unified training methodology in the construction sector.
- Have more qualified workers.
- Reduce the accident rate in the construction sector.

In this project, Industrial Association Groups (CNC, ANMOPYC, ANCE, ZSPS, AECOPS, AEP and VDMA), three SMEs (ENGIARTE, ROMANA SCAVI and LEGARRA) and three RTD performers (OKTAL, FHG-IGD and IKERLAN) will integrate their knowledge and experience to achieve the desired results.

The modules that are part of the PC-based simulator and Integrated Platform simulator can be summarised as follows:
- Educational and Training modules.
- Visualisation modules.
- Operator Interface modules.
- Dynamic Model modules.

IKERLAN, apart from being the project co-ordinator, is in charge of the design and development of the operator interface modules and dynamic model modules.

The Dynamic Model module is one of the key elements of the simulators. On one hand it has to react to the actions of the trainee on the operator interface modules. On the other hand it has to provide real-time information about the machine’s dynamics (mobile element position, speed and acceleration) to other modules involved in the simulation (visualisation or pedagogical modules, for instance) (see Figure 1).

The complexity of the dynamic model development arises from two requirements of the project:
- Realism: The trainee needs to experience a realistic sensation when working with the simulated machines. That is to say that the dynamic model has to accurately calculate the evolution of dynamic variables of the machine over time.
- Efficiency: The code corresponding to the dynamic model has to run on a standard low cost platform (desktop computer).

Taking into account the complex construction machines involved in the project, these two requirements are difficult to meet at the same time and are a challenge for the project team.

DESCRIPTION OF THE CONSTRUCTION MACHINERY SELECTED

As mentioned before, the four machines selected to be modelled within the project span the three large machinery groups that form the basis for the construction sector: Load lifting equipment, heavy works machinery and aerial work platforms. This implies that the selected machines are all very different from each other. Furthermore, each machine is a multi-domain system insofar as each of them contains a different kind of subsystems. Working on dynamic models of construction machinery therefore involves dealing with several fields of science such as 3-dimensional mechanics, hydraulics, electrical elements and control.
Hydraulic wheeled excavator

The hydraulic excavator, shown in Figure 2, is a construction machine principally used for digging trenches for cables and pipelines, as well as excavating foundations for new buildings. A hydraulic excavator basically consists of the following elements: an undercarriage where the wheels, stabilizers and dozer blade are located; a rotating superstructure with the diesel motor; a cabin containing all the operator commands for travel and digging; and a jib. The jib includes the boom, the stick and the end equipment. This final equipment could be a bucket, a clamshell or another item, although only the bucket has been considered here. This type of machine usually includes six hydraulic actuators, four cylinders and two hydraulic motors that deal with the locomotion and jib movements.

Dumper

A dumper (Figure 3) is a machine designed for horizontal transporting of light material such as soil or debris. Although its natural location is the construction site, this kind of machine can also be seen in other activity sectors like gardening and cleaning. The principal characteristic of the dumper is the small tilting tank (hopper) situated on the front part and provided with a frontal hydraulic unloading system. The driving seat is normally sited behind the hopper. The dumper can have four-wheel drive in addition to front-wheel drive, but the guiding wheels are always the front ones. Almost all dumper models have four gears and a lever for direction selection.

Work goods lift

A work goods lift is a machine consisting of a basket that slides up and down a lateral rail or two parallel rigid guides between the different floors of a building, as can be seen in Figure 4. This movement is attained when the motors located in the basket drive the pinions, which engage the rack of the guide. The work goods lift can only be used for lifting material, never for carrying people. The speed range of the machine is 10-25 m/min, depending on the load capacity of the model.

Mast climbing platform

The machine shown in Figure 5 is a mast climbing platform, the final machine chosen to be simulated within the scope of the VAR-TRAINER project. This machine consists of one or more vertical columns supported by a base, and a horizontal working platform. This machine moves in the same way as the work goods lift does, but there are some differences: the mast climbing platform is only for carrying people and the operator must be inside the platform. Moreover, as these machines are designed to allow workers easy access to the whole facade of a building, they move continuously across the building instead of moving from floor to floor.

Requirements of the dynamic models

The main aim of the project is to develop a versatile
simulator to train workers in the safe use of four different machines. The core of this simulator is a dynamic model that has to be able to run in real time on an ordinary PC as well as on a more powerful computer in the case of the Integrated Platform version. The real-time simulation implies that the system of equations describing the behaviour of the physical system must be solved analytically at fixed time instants (Lorenzo et al. 2001).

This requirement imposes a major restriction on the modelling process, and although a very complex model could be developed taking into account the exact behaviour of all the different components of the machines, some simplifications should be made with the purpose of building a real-time oriented model. As the simulator is conceived as a pedagogic tool, the components that do not have a critical influence on situations involving safety aspects may be modelled in a simpler way.

Therefore, this is the reason why the modelling activity focused on reproducing the reactions of the machines when a worker is operating them, in order to achieve the most realistic sensations and most believable risk situations.

**DEVELOPMENT METHODOLOGY FOR REAL-TIME DYNAMIC MODELS**

In the last years many commercial modelling tools have been tested at IKERLAN’s Control Engineering and Mechanical Engineering departments, according to their suitability, for the development of different mechatronic system dynamic models to be used in HIL simulations. These programs might be classified as: object-oriented modelling and simulation packages (Dymola), tools for modelling based on bond-graph techniques (20-SIM) and other specific programs for modelling mechanical systems (ADAMS, Pro-Mechanica) (Lorenzo et al. 2001; Otaduy et al. 2000).

Instead of using commercial software, there is another option for developing a dynamic model, which implies manual formulation and building of the equations that describe the behaviour of all elements of the modelled system.

Finally, and based on previous experiences, the commercial tool named Dymola was selected as being the most efficient way nowadays of building the core of the mathematical model of a 3-dimensional multi-domain system, due to the quality of the generated code and the ability to resolve algebraic loops (Landaluze et al. 2004). In any case, handmade blocks and objects would be added to the model built with standard Dymola objects in order to complete it and to manage the communications with the rest of the modules that constitute the simulator.

**Modelling in Dymola/Modelica**

Figure 6 shows the framework of the methodology followed to create the code of the dynamic models of the four selected machines. According to this framework, the modelling process begins using Dymola’s object oriented capabilities. Dymola supports hierarchical model composition, and libraries of reusable components are available in many engineering domains. All components contained in the libraries are described in Modelica, which is an object-oriented language for modelling almost any kind of multi-domain physical system. As in Modelica no particular variable needs to be solved manually, Dymola could be used to build a really large and complex model. Furthermore, Modelica supports several formalisms for writing the equations that describe the response of the elements: ODE (ordinary differential equations), DAE (differential-algebraic equations), finite state automata, etc. (Dynsim AB 2002).

A Dymola model is constructed on the basis of icons and connectors enabling a user-friendly graphical representation of the physical model. The model can be easily divided into submodels to simplify the verification and error-debugging phase.

Even if the modelling effort is significantly reduced when objects from standard libraries are used, Dymola offers the possibility to create new objects written in Modelica, grouping them into user-defined libraries.

Once all the elements that compose the model, the standard ones and the objects created by the developer, are properly connected, an integration method has to be selected before translating the model, taking into account that it must be run in real time.

The integration algorithm must be carefully selected. Usually the Explicit Euler integration method is used, but this algorithm is not suitable for real-time simulation because it imposes that an integration step size must be set shorter than the smallest time constant used in the model. The Implicit Euler method avoids this problem by allowing larger step sizes, but in turn it implies that a very large nonlinear system of equations needs to be solved at every step, which forces an excessively large step size to be set. In this situation the accuracy obtained could be unacceptable. To transcribe this kind of model Dymola offers new inline integration methods, where discretization formulas of the integration method are combined with the model equations and structural analysis and computer algebra methods are
applied on the augmented system of equations (Elmquist et al. 2003).

Within these inline integration methods the one called Mixed In-line Explicit/Implicit Euler has been chosen to translate and simulate the dynamic models of the four different machines.

As mentioned before, one of the requirements is that the simulator can be executed on a desktop computer. This means that, even if the proper integration method is selected, some simplifications might be carried out. In such cases Dymola should be used as a powerful simulation tool to test and check the accuracy of the simplified model.

**Integration Framework: Simulink**

During next step of the process, Simulink, from MathWorks, Inc., is used as an integration environment. Dymola’s Simulink Interface enables encapsulation of the code obtained from the translation of the Modelica model into a S-Function block, and therefore this block can be inserted in an ordinary Simulink model. This Simulink model can be completed by adding other hand-made S-Functions (as well as any block contained in Simulink block libraries), with the aim of implementing the pre-processing of the data the dynamic model receives from other components of the simulator and the post-processing of the information that the dynamic model module has to send to these other modules. At this point, some simulations should be done so as to check the stability and accuracy of the final model.

The last step of the procedure consists of obtaining the C-code representation of the final model. The tool named Real Time Workshop (RTW) is used for this purpose.

RTW is the foundation for Simulink code generation. It automatically generates stand-alone ANSI/ISO-C compliant C code for developing and testing systems modelled in Simulink. The resulting code can be used for many real-time and non-real-time applications. Code generation procedure can be carried out for an entire model or for an individual subsystem, enabling the code to run on any microprocessor or real-time operating system (The Mathwork Inc 2004).

The source code files resulting from the generation process are the base from which the final dynamic model module is built within the VAR-TRAINER development framework.

**DESCRIPTION OF THE EXCAVATOR MODELLING**

As has already been mentioned, working on dynamic models of construction machinery involves the use of multi-domain modelling techniques. Luckily, Dymola contains libraries that deal with all the necessary domains, as well as elements for linking different environments like mechanics and hydraulics.

![Figure 7: Multibody components of the excavator model](image)

**Modelling 3-dimensional structures**

The MultiBody library, a Modelica library for the modelling and simulation of 3-dimensional mechanical systems, has been used with the aim of creating the mechanical structure of the machines. As the main feature it can be mentioned that the developer does not have to worry about the kinematic loops that could appear in the modelling process, as occurs with the excavator bucket. In these cases Dymola uses an algorithm to transform overdetermined DAE equations (Differential Algebraic Equations), solving them analytically.

A set of MultiBody library components are designed to be directly connected with objects from other libraries such as one-dimensional translational and rotational mechanics libraries, hydraulic library or powertrain library, which makes the multi-domain modelling possible.

Another feature is the possibility of 3D visualisation of the evolution of the multibody components during the simulation, which is very helpful during the first steps of the modelling process (see Figure 8) (Otter et al. 2003).

As an example, Figure 7 shows the part of the model that composes the 3-dimensional structure of a hydraulic excavator.

Taking into account the low accelerations suffered, the necessity of developing a real-time algorithm (low computational cost) and the dimensions of the excavator, nonlinearities such as lateral play at joints, backlash and flexibility have been discarded. The excavator is modelled as a 3D rigid multibody system with perfect joints. Each part is represented with a Body element that defines its mass and inertia.

Actuated revolute joint objects are used to define the rotatons along different axes. These objects have an additional flange connection. Thus, different elements of the
rotational library can be attached in order to drive the revolute joints.

For the modelling of the connection between the stick and the bucket, the Multibody library offers a special component consisting of 3 revolute joints (RRR), which when connected to the stick form a planar loop. Dymola takes charge of solving the resulting non-linear system of equations (Beater and Otter 2003).

There are other objects attached to the bodies that describe a line force interaction between the attachment points. The magnitude of this force depends on the actuator element attached to the translational connectors.

The Visualisation module, one of the modules making up the simulator as a whole (Figure 1), is capable of detecting collisions between any of the four machines and any surrounding element sited in the simulation scene. When a collision occurs, the Dynamic Model module is informed of the location and penetration of this collision. A force proportional to the penetration is calculated and sent to the model using the force input elements attached to each body of the machine.

**Modelling hydraulic systems**

Both the excavator and the dumper have hydraulic circuits with hydraulic actuators for producing mechanical movements of the mobile parts of the machines. The overall drive system of an excavator is composed on the one hand of a diesel engine and a drive train, and on the other hand of a hydraulic circuit. The hydraulic circuit contains two pumps, two motors, four cylinders and several valves. The hydraulic system of the dumper is much more simple. In this case only a pump, a cylinder and a valve need to be modelled.

Despite the fact that, due to real-time requirements, real hydraulic circuits were not modelled exactly, the resulting movements are extremely realistic. Figure 9 shows parts of the hydraulic system of the wheeled excavator attached to the previously presented mechanical structure.

The circuit of each actuator has been modelled separately from the others, only taking into account the actuator, a proportional valve, an externally commanded flow source and a tank element. These circuits have been mainly modelled with components belonging to Dymola’s HyLib library. As regards the modelling of the most complex actuators, a double acting hydraulic cylinder and motors with leakage were selected. Valves and pumps were also modelled using appropriate elements from the HyLib library.

In the figure only the hydraulic components for the boom and the hydraulic motor driving cabin swing are shown. As can be seen, the cylinder of the boom is directly attached to the mechanical part, connecting it to the line force element in order to exert the force that rotates the axis of the boom. The chambers of the cylinder are connected to a proportional valve with second order spool dynamics and nonlinearities. This valve, externally commanded by the operator, drives the flow injected by the flow source to one of the chambers connecting the other to the tank. The circuits developed for the stick and the bucket contain the same elements but have different parameter values.

The constant displacement hydraulic motor that generates the power for swinging the upper structure is connected to the carriage revolute joint via an ideal gear. The rest of the elements are the same as described above: a proportional valve and a flow source.

The cil that the flow sources need to insert into each actuator is computed once the model is encapsulated in Simulink via an S-Function, and it depends on the pedals, joysticks and levers controlled by the operator, the flow request from the other actuators and the speed of the diesel engine. In turn, the torque, and consequently the speed, of the diesel engine modelled in Simulink is a function of the pressure differences of all the actuators.
The function of the other hydraulic motor is to propel the machine, and like the other actuators it is controlled by a valve, which uses the oil from a flow source (Figure 10). The mechanical connector of this variable displacement motor is connected to a coulomb friction-based brake. After the brake, there is a gear with mesh efficiency and bearing friction, followed by a rotational component that represents the inertia of the wheel. The locomotion system is completed with an object specifically developed for this simulator. This object, which will be explained below, deals with the interaction between the wheels and the ground.

Modelling ground-wheels interaction

For the excavator and dumper, a key point in the dynamic behaviour of the machines are the forces between the wheels and the ground. Some of the most risky operations, rocking and sliding, are related to these forces.

The interaction forces between the soil and the wheels are modelled taking the following into consideration:

- The virtual machine should move following the ground contours.
- It should allow loss of stability to be emulated (sliding and rocking).
- Traction and braking movements should be allowed.
- Contact between the stabilizers and the soil must be allowed during the working phase (for the excavator model only).

Two different phases are taken into account: the working phase and the displacement phase. In both cases the interaction between the soil and the machine is modelled as a visco-elastic force.

Displacement Phase:

This phase is normally used to move from one working area to another or travel along a road. The normal reaction, which represents the movement of the machine along the virtual surface, is computed proportional to the penetration of the wheel in the ground (for the elastic part) and proportional to the penetration rate (for the viscous part) to damp the vibration and aid the stability of the model.

Working Phase:

The working phase is characterised by no relative displacement of the contact point between the wheel and the ground. Both parts of the force take into account the contact stiffness of the tyre and the ground, calculated as two springs and two dampers in series (Figure 11).

This is a non-linear model allowing contact loss of the wheel when no penetration is detected. The penetration value is computed at the point where the distance between the centre of the wheel and the ground is minimum.

A user model has been developed to simulate the contact forces between the wheel and the ground. The circular icon in Figure 10 represents the developed object. This model takes into account the traction and lateral forces exerted between the wheel and the ground based on the friction between the two elements, but with different approaches.

The lateral direction is computed in a simple way: its value is computed to avoid lateral displacement (PI controller) until the force exceeds the resultant of the maximum friction force ($\mu N$), where $N$ is the penetration force explained above and $\mu$ the friction coefficient. When the maximum friction force is reached the machine starts to slip laterally, entering a dangerous situation.

The forward direction is computed in a more realistic way using a variable friction coefficient $\mu_{\text{slip}}$, which depends on the relative sliding between the contact point of the wheel and the ground (Bakker et al 1989). The friction coefficient is zero when there is no relative sliding (Vrel/Vmax), and shows a maximum at a certain value of the relative sliding (when accelerating or braking), normally around 20% of the relative sliding. The shape of the function depends on three parameters (B, C and D) (see Figure 12) that can be tuned to modify the shape of the friction value. We have chosen a set of values for the simulation to reach the maximum value when sliding is equal to 1 and 95% of the maximum friction when sliding is 0.2. This allows a more stable simulation when a large integration step is used. When the time integration requirement is not so critical a more realistic set of parameters is used.

This enables a more realistic simulation of the sliding phase and a smoother change of the forces at the beginning and end of the displacement phase.
Figure 12: Friction coefficient shape for forward direction ground. The forces are applied to the contact points with the ground, wheels or stabilizers, depending on the machine type.

In this case, the lateral and forward direction forces are computed in the same way lateral force is computed in the displacement phase. In this case the penetration is computed at the end of the stabilizers.

Modelling ground-bucket interaction

In an excavator, the interaction between the ground and the bucket is a very complex phenomenon depending on a large number of parameters: cutting effect, ground shearing, filling force and interaction with solid material such as rocks. Some authors have proposed very precise models based on material science, but the computational requirements are too great for a real-time simulation on a desktop PC (Lorenzo et al. 2001).

In order to meet the computational constraints a simplified model proposed by Luengo et al. (1998) has been used. This model takes into account the influence of the soil weight, cohesion of the soil, surcharge pressure and the ground slope. Friction forces are neglected. All the geometrical parameters are computed from the Visualisation module.

CONCLUSIONS

The work carried out during the project corroborates the suitability of the development methodology followed, especially due to the open nature of this methodology. Although it is based on commercial tools like Dymola and Simulink, these tools allow the inclusion of hand-made objects and blocks, building mixed models where standard objects and blocks from commercial libraries coexist in the same model with parts specially developed for the project. For instance, the presented ground-wheels and ground-bucket interaction models.

The selected modelling methodology has also worked well. Hierarchically composed modelling, with different separately modelled systems, makes it possible for any part of the models built to be reused in a possible future development of other machine dynamic models.

The hardest task was to achieve suitable models to be simulated in real time on a desktop computer. On the one hand, the model must be simplified with the aim of reducing the number of equations that describe the model mathematically while maintaining the desired accuracy. As a result the computational cost decreases, enabling the reduction of the step size. On the other hand the model must be set up changing the value of several parameters, so that it can be run with the highest step size without losing stability. A higher step size implies that the model could run in real time on a less powerful PC.

For instance, in the case of the excavator, the step size had to be set to 10 ms in order to solve in real time more than 5800 DAE equations that represent the developed dynamic model module on a computer equipped with a Pentium 4 processor running at 2.4 GHz, with the rest of the modules executing at the same time.

ACKNOWLEDGMENT

The material in this paper has been partially funded by the European Union under the Collective project “VAR-TRAINER” (EC-Contract n° COLL-CT-2003-500452).

REFERENCES


THE NEED FOR A SIMULATOR IN DUTCH DRINKING WATER TREATMENT

Ignaz Worm
Duinwaterbedrijf Zuid-Holland
PO Box 34
2270 AA Voorburg
The Netherlands
i.worm@dzh.nl

Luuk Rietveld
Delft University of Technology
Faculty of Civil Engineering and geosciences
PO Box 5048
2600 GA Delft
The Netherlands
i.c.rietveld@tudelft.nl

KEYWORDS
Industrial control, production, training, decision support system, dynamic modelling.

ABSTRACT
Encouraged by the need to increase efficiency, the daily operation of Dutch drinking water treatment plants is being automated and centralized. By doing so a conflict is introduced. The function of the traditional operator will soon be replaced with a supervisor who has a larger span of control but less natural training, since he takes no longer part in daily operation. Duinwaterbedrijf Zuid-Holland acknowledges this problem and develops a simulator for supervisors to be trained, tested and supported during calamities. The responsibilities of the supervisors are the basis for the development of the simulator.

In the Netherlands, the annual drinking water production measures approximately 1150 million cubic meters. The water is treated and distributed by fourteen water supply companies, using surface water, ground water and infiltrated river water as sources for drinking water, see figure 1. Typically, Dutch drinking water is disinfected without the use of chlorine but with a multi barrier system against pathogenic micro organisms and organic micro pollutants. Recontamination is prevented and the water has a low potential for regrowth. The water is softened for comfort, to reduce the use of detergents and to prevent scaling of household equipment.

Duinwaterbedrijf Zuid-Holland

With an annual production of 75 million cubic meters, Duinwaterbedrijf Zuid-Holland is an average sized water supply company. The company uses infiltrated river water as raw material that is treated with softening, aeration, dosing of activated carbon, rapid sand filtration and slow sand filtration, see figure 2. The dune water is recharged with pretreated water from the river Meuse. Pretreatment consists of iron sulphate dosing, microsieving, iron chloride dosing and rapid sand filtration. Most of the water is infiltrated via open infiltration lakes in the dunes.

Figure 1: Dutch water supply companies (source: VEWIN Water supply statistics 2004)

Figure 2: Artist impression water treatment at Duinwaterbedrijf Zuid-Holland

Present operation
Drinking water treatment processes are typically continuous, with almost no shut-downs and start-ups. Not only the average water quality and pressure in the network is important, the prevention of incidents as well. Sickness of one child, man of woman as a consequence of
consumption of drinking water can do serious damage to the image of the water supply company. Typically concentrations of elements in drinking water are low.

The largest of the three water treatment plants of Duinwaterbedrijf Zuid-Holland, Scheveningen, has a one man 24/7 shift. This operator decides on the daily volume of water that is produced and on the levels of drinking water reservoirs, on the treatment plant as well as in the distribution network. If he wants to, he can use a demand prediction model, developed by one of the operators, in Excel. The model needs several parameters to be filled out manually to work properly. For adjustment of the production flow, usually he relies on his experience and the development of the demand for drinking water during his watch. He is responsible for the pressure in the distribution network. He reacts to pressure, water quality or water level changes by switching pumps on and off, or by changing dosage of chemicals. Parts of the plant are automated such as the backwashing of the rapid sand filters and the production of calcium hydroxide for the softening process.

FULLY AUTOMATED OPERATION

To increase efficiency, on January 1st 2009 the daily operation of Duinwaterbedrijf Zuid-Holland will be fully automated. Objective is to ban human interference as much as possible. In 99 percent of the instances the intake of river water, pre-treatment, infiltration, treatment and distribution should take place automatically. To face this challenge Duinwaterbedrijf Zuid-Holland and PWN Waterleidingbedrijf Noord-Holland joined forces and developed a hands-off automation and operation concept. At PWN Waterleidingbedrijf Noord-Holland this concept is operational since July 2005. The operation philosophy is based on a daily prediction of demand and volume setpoints computing for the production plants twice a day on fixed times. This will lead to optimal use of the installed production capacity, delaying the need for (re-) investments, a more stable water quality and minimal use of energy and chemicals.

Supervisor

With the introduction of fully automated operation the job position of operator will be replaced by a supervisor. The distinct difference is that the supervisor will be responsible for the operation of five plants in stead of one and for the water distribution. During normal working hours he will validate production data and analyse deviations. In shifts, the supervisor stays responsible for dealing with emergencies, alarms and for “long distance” problem solving, 24/7. To be competent for both types of tasks, he needs to understand the entire treatment system thoroughly. He needs to speak the language of automation fluently and not only have the knowledge but also the skills to be able to react adequately in the one percent non regular operation situations.

Responsibilities

To investigate the impact of the changes in responsibilities of the (future) supervisors questionnaires were filled out by nineteen people at Duinwaterbedrijf Zuid-Holland and PWN Waterleidingbedrijf Noord-Holland. Special attention was paid to the differences in perception of responsibilities between supervisors and managers and between the two companies.

In general no large differences were identified in the perception of responsibilities between supervisors and managers. PWN Waterleidingbedrijf Noord Holland mentions the importance of team spirit and communication skills, employees of Duinwaterbedrijf Zuid-Holland emphasize that manual operation should be prevented. Both managers and supervisors of PWN Waterleidingbedrijf Noord-Holland describe the responsibilities of the supervisor as being responsible for daily operation of drinking water treatment from source to tap.

The overall response showed that the supervisor is to be responsible for the water delivery from source to transport mains, in terms of pressure, quantity and water quality. This main responsibility is further specified. The supervisor is responsible for a minimal pressure of 25 m above ground level in transportation mains. Pressure changes have a typical time scale of milliseconds to minutes. Therefore the responsibilities of the supervisor in terms of pressure have a timescale of minutes as well. For quantity a time scale of 24 hours is used. This means the supervisor is responsible for proper functioning of demand prediction and setpoint computing. For water quality the supervisor needs to react when quality parameters deviate beyond limits of operational windows and when incidents are reported by the outside or by the laboratory. A typical timescale is hours to 24 hours. It was concluded that the number-one-fear of managers as well as of supervisors is lack of knowledge of the operation.

Conflict

One might think that fully automated treatment plants will require less sophisticated operator care, but the opposite is true. Operators will need to become more sophisticated and able to understand what is happening behind the “treatment plant chatter” (Trussell, 2000). Education requirements will rise and high-powered computer programs will assist the supervisors. Duinwaterbedrijf Zuid-Holland is the first water supply company to identify the question how its supervisors can perform their new larger and more complex task and deliver the same quality as the present operators. The supervisor will lack natural training of the daily operation, will need to know the ins and outs of five treatment plants in stead of one, and will need to adopt a completely new automation structure.
SIMULATOR

Duinwaterbedrijf Zuid-Holland acknowledged the conflict and decided to take all measures needed to provide the supervisor with initial knowledge and skills for his new task and to maintain this knowledge and skills in the future. Therefore a simulator is needed. In this simulator supervisors will be able to train, be tested and be supported in making decisions during calamities. Ultimately, the simulator can be used for process optimization and for the prediction of calamities. The justification for the development of a simulator lies in the responsibilities supervisors are to carry.

The simulator will lead to a more thorough understanding of the processes and to a more critical attitude to the process automation. For training and testing both imaginary and historical cases can be exercised. For support during a calamity predefined procedures will be presented as well as the way a calamity was dealt with by other supervisors earlier.

The use of simulators is common in the process industry. In Finland a simulation environment has been developed for paper and board mills, for the purpose of process and automation design, analysis and operator training (Niemennaa et al, 1998). The environment consists of standard unit operation models based on the first principles of physics and chemistry whenever possible. The modules are standard building blocks, including automation and electrical systems. The user can simulate his own paper mill by using these blocks. The simulator is either connected with a program emulating control room display or to a real interface of automation system. Use of the simulator has lead to changes in operation parameters. The use of the simulator for training has been discussed (Lappalainen et al, 2001).

German power supply is facing rapid changes in system operation because of diverse non technical impacts, requiring extensive coordination and parallel acting of all control centres involved under normal as well as abnormal operating conditions. Training the operator’s decision making and, in doing so, keeping his knowledge, skills and experience up to date in theory as well as in practice is the key factor of today’s German power system operation (Spanel et al, 2001). The simulator which has been developed has different man machine interfaces for operators of the “power pool” and the regional control centres, and for the trainer. Models in the simulator are key factors for the success of practical training, meaning that they must be representative for normal as well as disturbed operation situations. To keep the overall effort within tolerable limits, a uniform and easy accessible database of production data must be available. The simulator has been implemented in both Germany and the Netherlands. For training purposes, basic scenarios are available and can be arbitrarily modified.

Models

In future, models will be constantly run to predict water quality and to determine the necessary treatment. This will lead to a better water quality, while minimizing the use of chemicals (Rosen, 2000). Models will be an essential part of the simulator as well. Real sensors and actuators must be replaced by object models and their behaviour must be simulated using simulation models. The object models will be the same ones as used in the factory acceptance tests of the automation system. Using object and simulation models, the effects of human interference in operations can be monitored offline, without disturbing the treatment and other processes.

A model is a schematic representation of a part of a real life situation to serve a certain objective (Battjes, 1999). To use and or to develop models, following this definition, real life situations must be identified “serving a certain objective”. These objectives are derived from the responsibilities of the supervisors. Separate models will be used for pressure, quantity and water quality, each with its own time scale.

Pilot

UREsson, ABB, Delft University of Technology, DHV Engineers and Duinwaterbedrijf Zuid-Holland, realize a pilot in which a small simulator is built. The objective is to develop a simulator for one treatment step of one of the production locations. The softening of water treatment plant Monster has been chosen. A copy of the treatment step automation software is connected with the decision and simulation support tool USE and USE is connected to the Simela water quality model for softening, see figure 3. USE is a commercial programming environment in which intelligent applications can be created for automation of fault detection and diagnosis, operational predictions, operator advisories and workflow management. It enables domain experts such as operators, process engineers and quality control managers to specify their lines of reasoning concerning production problems and quality issues. The decision and simulation support tool will have the same topology as the automation system of the softening plant and will show the water flows, the information flows and the power status. As mentioned drinking water treatment processes have different typical time scales. To be able to train, test and support in an efficient way, in USE the ratio between real time and simulation time is set and distributed to the object and simulation model.

Stimela is a modelling environment for drinking water quality developed by Delft University of Technology and DHV Engineers [van der Helm, Rietveld, 2002]. The Matlab/Simulink models are dynamic, which is essential for application this pilot, open source and freely accessible via internet. Within Stimela for each treatment process a model has been developed. In the pilot the model for softening using a fluidised bed reactor is applied,
calculating the hardness of water as a function of design and operation parameters such as water flow, bypass percentage, incoming hardness, caustic soda flow and pellet discharge speed.

For adoption by its users it is essential that the simulator behaves and looks like the real automation system. Therefore the user interface of the trainee has the “look and feel” of the automation system. The user interface of the trainer will have a tailor made “look and feel”.

Duinwaterbedrijf Zuid-Holland defined a use case for this pilot, mentioning two cases of which one calamity. In a data model the input has been described which is needed for initiation of the training case and for running the simulation. This initiation consists of values for the object model and values for the Stimela simulation model, including the ratio between real time and simulation time. Furthermore in the use case the requested output of the simulation model has been described, the preferred operator handlings and the requirements for the report of the training.

The pilot will be completed with a business case for company wide introduction.

FUTURE RESEARCH

When the pilot has been successful and company wide application of the simulator is realised, the existing network pressure, water quantity and the remaining Stimela water quality models will be integrated and validated. Study must be carried out to determine the didactic optimum between training in the simulator and in the field, as well as the most effective way of testing. Finally more use cases, in particularly for calamities, will be defined.

CONCLUSION

Duinwaterbedrijf Zuid-Holland acknowledges the problem the supervisor can face in a fully automated hands-off operation of its drinking water treatment plants. Responsibilities of the supervisor have been identified to form the basis for the development of a simulator. The simulator will be used for training, testing and supporting of supervisors. In a pilot two cases will be evaluated of which one will be a calamity.

REFERENCES


Figure 3: Architecture of the simulator
PHYSICAL RESERVOIR AND WATER FLOOD SIMULATOR OF THE MIJARES RIVER

Nicolas Montés, Leopoldo Armesto, Josep Tornero
Dpto. Ingeniería de Sistemas y Automática
Technical University of Valencia
46022 Valencia (Spain)
e-mail: nimensan@upvnet.upv.es, leorara@isa.upv.es, jtornero@isa.upv.es

ABSTRACT
This paper describes an adopted solution to study of the Mijares River (Valencia, Spain) when the flood water collides with the reservoir water. In our approach, a real physical scaled model (1:75) has been constructed in order to reproduce the desired conditions for the study, based on preliminary results. For this purpose, simulators of the flood water and reservoir are required, which are modeled as a complex system of coupled tanks. Robustness and precision are required for the application, since the scale magnifies the errors. Results show the accuracy of the proposed control system, where multi-rate techniques have been used in order to improve the performance of the control system. In addition, a SCADA interface based on GEMMA state machine has been developed. This software is used for monitoring and control and is able to deal with system failures.

KEYWORDS
Water flood, reservoir, multirate, physical simulator, control architecture, EKF

1. INTRODUCTION
In the Mediterranean coast, there is a physical phenomenon named as “cut-off low”. This phenomenon is produced because of high sea water and cool air temperatures, which causes fast evaporation and condensation. Typically, these specific conditions, are given in autumn, causing strong rains in very short time periods at the surroundings of the seashore. For instance, in 1987 in Valencia (Spain), a cut-off low left 1000 l/m² of rain in less than 36 hours. Under these conditions, rivers increase drastically their volume and reservoirs must be ready for bearing this huge volume of water. However, there is another interesting phenomenon related to flood water when collides with reservoir water, which generates a wave of rushing water. This collision may cause overflow at the nearby of the wave and also the erosion of the reservoir floor (Steven et. al., 2003). The wave position, dimensions and the effects are unknown or very difficult to analyze a priori. In that sense, in order to study its behavior, it is necessary to simulate and reproduce a scaled model of the flood water, the reservoir and the relief and roughness of the terrain.

(Bozkus and Kasap, 1997) try to solve a similar problem, obtaining a flood area in the case of dam breaks-down. In this work, some numerical Dam-flood models are compared with physical scaled model. Their conclusion is that it exist large differences between the physical model and the numerical model predictions, being numerical models inappropriate for their application. (Steven et. al., 2003) develop a physical scaled model of a dam in order to introduce structural changes on the dam to reduce erosion and downstream dissolved gas concentrations.

This paper focuses on the simulation and reproduction of a scaled model of the Mijares River, in Valencia (Millan et. al., 1997),(CEAM, 1998). Preliminary studies developed by our partners from Department of Hydraulic Engineering at the Technical University of Valencia determine the standard behavior of the Mijares River and his effluents in case floodwater with a return 500 years period. These results provide the conditions to be reproduced on a real physical scaled model simulator. In particular, the flow of the floodwater and the behavior of the reservoir are given. These preliminary studies are described in Section 2, also in this section is explained a physical scaled simulator. In Section 3 is described water flood simulator and Section 4 describes the reservoir simulator. A SCADA software supervises and controls the overall application, which is described in Section 5. Section 6 describes the hardware architecture based on industrial components and some conclusions are given in Section 7.
2. PRELIMINARY STUDIES

As hereinbefore commented, the water flow of Mijares River and its effluents in a standard floodwater situation with a return period of 500 years is considered for the simulation. This flow is depicted in figure 1 (continuous line), where it can be appreciated that it presents a peak, which is the most critical time instant. In addition to this, the height (above the sea level) of the reservoir related to this water flow is also depicted in figure 1. In particular, three different initial heights have been considered: with the reservoir nearly empty (height 525 m., dash-dotted line), a normal situation (height 560 m., dotted line) and close to overflow situation (height 580 m., dashed line).

![Figure 1 Flood Water](image)

Figure 1 Flood Water

In order to know the real effects on the river sides caused by the floods, a scaled model is required. This model is sensory in order to know in real-time the position and transition of the wave caused.

![Figure 2 Robot Arm](image)

Figure 2 Robot Arm

Numerical models fail when trying to simulate this kind of effects. Therefore, a scaled physical relief must be used instead of the numerical.

Usually the holography is reproduced by hand, so many human aim are committed with a task of high time consuming. In own case, a robot setup has been involved in this tedious task as shown in Figure 2. Specific CAD/CAM software translate 3D CAD model into robot TCP coordinates with very high precision.

In order to reproduce such conditions, a scaled physical model has been built with a reducing factor of 1:75. Figure 2 is a schematic representation of this model, where three separated systems can be clearly seen: the flood simulator, the physical relief of the narrowing and the reservoir simulator.

A constant flow $Q_v$ is introduced in the flood simulator (right side of the figure). This flow can be regulated with a V-shape gate that modifies the desired water flow to be introduced in the narrowing. Based on the expected flow peak on the real situation, the maximum flow of the scaled model corresponds approximately to 58 l/sec., which is the nominal (constant) value of a water bomb located at the beginning of the model. In order to perform flow regulation, a sonar sensor has been placed at the end of the flood simulator, on a deposit which height is directly related with desired flow to be reproduced. In addition to this, two holes at the beside the flood gate are used to evacuate the surplus of water. The dynamic model of this flood simulator is described in Section 3.

The reservoir begins when the narrowing becomes wider (left side of figure 3). This part has been implemented with a single deposit, where a sonar sensor is used to measure its height and a gate regulates the flow of water to evacuate. The goal of this part is to reproduce the height of the reservoir as given in figure 1 for different height levels at their corresponding scaled values.

![Figure 3 Physical scaled simulator](image)

Figure 3 Physical scaled simulator
3. WATER FLOOD SIMULATOR
DYNAMIC MODEL

Figure 4 Physical flood model

The detailed description of the flood simulator can be found in figure 4. The constant flow \( Q_v \) acts as a known perturbation of the system. The V-shape gate is controlled with a stepper motor and can be modelled as two complementary valves, one related to spillway tank T3 and the other related to tank T2, which is an open drain tank. Tank 4 receives as input the water from tank T2, although they are not directly connected as shown in figure 5. Tank T5 is coupled with tank T4 and a sonar sensor is used to measure its height, which is directly related to the desired water flood. The sink of T5 has a triangular-shape in order to increase the sensibility of the sensor.

Next figure shows the representation and coupling among as a simplified representation of the scale model. The purpose is to implement the appropriate control law for the valve “V”.

Figure 5 Dynamical model to study

The equations to represent dynamic behaviour are:

\[ h(t) = f[h(t), x(t), Q_v] \]

where \( h(t) = [h_1(t), h_2(t), h_3(t), h_4(t), h_5(t)]^T \) is the state vector composed of heights of tanks, \( Q_v(t) \) is the input flow and \( x(t) \) is the control input of the valve within the range \([0..1]\) (“0” means closed gate and “1” means open gate). In particular, the following relations can be deduced from figure 4:

\[
\begin{align*}
Q_{i1}(t) - Q_{i2}(t) - Q_{i3}(t) &= A_i \cdot h_i(t), \\
Q_{i3}(t) - Q_{i4}(t) &= A_i \cdot h_i(t), \\
Q_{i2}(t) - Q_{i5}(t) &= A_i \cdot h_i(t), \\
Q_{i5}(t) - Q_{i6}(t) &= A_i \cdot h_i(t)
\end{align*}
\]

where \( Q_{ij}(t) \), \( Q_j(t) \) are output flows; \( Q_{ij}(t) \) is the flow between tanks \( i \) and \( j \); \( A_i \) is the tank \( i \) area.

Moreover, the gate presents an important non-linearity (switching condition) related to the effective coupling for evacuating water from T1 to T2 and T3 or vice versa. In fact, when the valve is nearly closed \( x(t) \), only affects to T2, while the coupling between T1 and T3 has a constant maximum length of \( l_{13,\text{Max}} \). On the other hand, when the valve is open, \( x(t) \) only affects to T3 and the coupling between T1 and T2 is constant \( l_{12,\text{Max}} \). Therefore, the effective coupling between T1, T2 and T3 can be modelled as:

\[
\begin{align*}
l_{13}(t) &= \min(l_{13,\text{Max}} \cdot x(t) \cdot (l_{12,\text{Max}} - l_{13,\text{Max}})) \\
l_{12}(t) &= \min(l_{12,\text{Max}} \cdot (1 - x(t)) \cdot (l_{12,\text{Max}} - l_{13,\text{Max}}))
\end{align*}
\]

In our physical model, \( l_{12,\text{Max}} = l_{13,\text{Max}} \) therefore the switching condition is given when \( x(t) = 0.5 \).

In addition to this, because of the physical configuration the flow between T1-T2 and T1-T3 depends on the difference of heights of the tanks. The flow equations are:

\[
\begin{align*}
Q_{i1}(t) &= C_{D1} \cdot l_{13}(t) \cdot (h_i(t) - h_j(t)) \cdot \sqrt{2 \cdot g \cdot (h_i(t) - h_j(t))} \\
Q_{i2}(t) &= C_{D2} \cdot l_{12,\text{Max}} \cdot h_j(t) \cdot \sqrt{2 \cdot g \cdot h_j(t)} \\
Q_{i3}(t) &= C_{D3} \cdot a_i \cdot \sqrt{2 \cdot g \cdot h_i(t)}
\end{align*}
\]

where \( C_{D1}, C_{D2}, C_{D3} \) are valve discharge coefficients.

Similarly, flow equations of tanks T4 and T5 can be modelled as follows:

\[
\begin{align*}
Q_{i4}(t) &= C_{D45} \cdot a_{i5} \cdot \sqrt{2 \cdot g \cdot (h_i(t) - h_j(t))} \\
Q_{i5}(t) &= 0.5785 \cdot \frac{8}{15} \cdot \sqrt{2 \cdot g \cdot \tan\left(\frac{\pi}{4}\right)} \cdot (h_i(t) + 0.0009)^{2.5}
\end{align*}
\]

where \( C_{D45} \) are valve discharge coefficient. Flow equation corresponds to specific calibration of the triangular-shape output valve of tank T5.
Figure 6 shows control architecture of the water flood simulator:

```
<table>
<thead>
<tr>
<th>Inverse Model</th>
<th>PID</th>
<th>Flood Simulator</th>
</tr>
</thead>
<tbody>
<tr>
<td>Error</td>
<td></td>
<td>Flood Reference</td>
</tr>
</tbody>
</table>
```

**Figure 6 Flood loop control**

Flood reference is used in order to compute the valve position based on an inverse model. In addition to this, a PID controller is used to improve the dynamic characteristics and robustness of the system (Ogata, 1994). A Kalman filter (Bar-Shalon, 2001) is used to estimate the water flood flow based on sonar measurements (height of tank T5). This observer is also used to estimate the state of the system, which is useful for monitoring the whole process, as described in Section 5.

3.1 INVERSE MODEL

The inverse model can be obtained by solving the following set of algebraic equations at stationary conditions:

\[ Q_s = Q_{s2} + Q_{s3}, \quad Q_{s3} = Q_0, \quad Q_{s2} = Q_{s4} = Q_{f}' \]

where \( Q_{f}' \) is the desired flow reference. The solution of these equations is:

\[
h'_f = h'_s + \sqrt{\frac{Q'_f}{C_{DF}^2 \cdot a_{DF}^2 \cdot 2 \cdot g}}
\]

\[
h'_s = h'_f + \sqrt{\frac{Q'_f}{C_{DF}^2 \cdot a_{DF}^2 \cdot 2 \cdot g}}
\]

\[
h'_s = \frac{(Q_s - Q'_f)^2}{C_{DF}^2 \cdot a_{DF}^2 \cdot 2 \cdot g}, \quad h'_s = \frac{(Q_s - Q'_f)^2}{C_{DF}^2 \cdot a_{DF}^2 \cdot 2 \cdot g}
\]

3.2 KALMAN OBSERVER

In order to obtain an estimation of heights and flows for all tanks, an Extended Kalman filter (EKF) is used (Bar-Shalon, 2001). For the implementation, a discrete-time dynamic model must be used. For simplicity, the following approximation has been considered:

\[
h_s \approx h_{k-1} + f(h_{k-1}, x_{k-1}, Q_{k}) \cdot T_s
\]

where \( T_s \) is the sampling time. The well-known equations of the EKF are:

\[
h_{k-1} = h_{k-1} + f(h_{k-1}, x_{k-1}, Q_{k}) \cdot T_s
\]

\[
P_{k-1} = A_D \cdot P_{k-1} \cdot A_D^T + B_D \cdot Q_s \cdot B_D^T
\]

\[
K_s = P_{k-1} \cdot C_D^T \cdot (C_D \cdot P_{k-1} \cdot C_D^T + R_k)^{-1}
\]

\[
h_k = h_{k-1} + K_s \cdot (h_{k-1} - C_D \cdot h_{k-1})
\]

\[
P_k = P_{k-1} - K_s \cdot C_D \cdot P_{k-1}
\]

where:

\[
A_D = I + \nabla f \cdot T_s, \quad B_D = \nabla f \cdot Q_s \cdot T_s, \quad C_D = [0, 0, 0, 1]
\]

And \( Q_k \) is the covariance matrix modeling uncertainties on the valve position \( x(t) \) and input flow \( Q_s \). \( R_k \) is the covariance matrix related with the sonar uncertainty.

In order to compute the Jacobians, the switching condition of valve “V” must be taken into account. In particular, for the stationary conditions when \( x' < 0.5 \)

\[
\frac{\partial Q_{f}'}{\partial h_s} = \frac{3}{2} \cdot C_{DF} \cdot x' \cdot (l_{2,M} + l_{3,M}) \cdot \sqrt{2 \cdot g \cdot (h'_s - h'_f)}
\]

\[
\frac{\partial Q_{f}'}{\partial h_s} = -\frac{3}{2} \cdot C_{DF} \cdot x' \cdot (l_{2,M} + l_{3,M}) \cdot \sqrt{2 \cdot g \cdot (h'_s - h'_f)}
\]

\[
\frac{\partial Q_{f}'}{\partial Q_s} = \frac{3}{2} \cdot C_{DF} \cdot (l_{2,M} + l_{3,M}) \cdot (h'_s - h'_f) \cdot \sqrt{2 \cdot g \cdot (h'_s - h'_f)}
\]

\[
\frac{\partial Q_{f}'}{\partial h_f} = \frac{3}{2} \cdot C_{DF} \cdot l_{3,M} \cdot \sqrt{2 \cdot g \cdot (h'_s - h'_f)}
\]

\[
\frac{\partial Q_{f}'}{\partial Q_s} = \frac{3}{2} \cdot C_{DF} \cdot l_{3,M} \cdot \sqrt{2 \cdot g \cdot (h'_s - h'_f)} \cdot \frac{\partial Q_s}{\partial Q_s} = 0
\]

Thus, when \( Q_f' < Q_{f,se} \), means that the valve must be nearly closed and therefore the height of tank T1 and the position of the valve can be computed as follows:

\[
Q_{f, se} = 19.3163 \text{ (l/s)}
\]
And on the other hand, when $x' > 0.5$

$$\frac{\partial Q_{12}}{\partial t} = \frac{3}{2} C_{DF} \cdot l_{12\text{Max}} \cdot \sqrt{2 \cdot g \cdot (h'_x - h'_y)}$$

$$\frac{\partial Q_{13}}{\partial h_h} = \frac{3}{2} C_{DF} \cdot l_{12\text{Max}} \cdot \sqrt{2 \cdot g \cdot (h'_x - h'_y)}$$

$$\frac{\partial Q_{13}}{\partial h_x} = \frac{3}{2} C_{DF} \cdot l_{12\text{Max}} \cdot \sqrt{2 \cdot g \cdot (h'_x - h'_y)}$$

$$\frac{\partial Q_{13}}{\partial h_y} = \frac{3}{2} C_{DF} \cdot l_{12\text{Max}} \cdot \sqrt{2 \cdot g \cdot (h'_x - h'_y)}$$

$$\frac{\partial Q_{13}}{\partial t} = C_{DF} \cdot (l_{12\text{Max}} + l_{13\text{Max}}) \cdot \sqrt{2 \cdot g \cdot (h'_x - h'_y)}$$

The rest of the partials derivatives are:

$$\frac{\partial Q_{21}}{\partial h_h} = \frac{3}{2} C_{D2} \cdot l_{12\text{Max}} \cdot \sqrt{2 \cdot g \cdot (h'_x - h'_y)}$$

$$\frac{\partial Q_{21}}{\partial h_x} = \frac{3}{2} C_{D2} \cdot l_{12\text{Max}} \cdot \sqrt{2 \cdot g \cdot (h'_x - h'_y)}$$

$$\frac{\partial Q_{21}}{\partial h_y} = C_{D2} \cdot a_{25} \cdot \sqrt{2 \cdot g \cdot (h'_x - h'_y)}$$

$$\frac{\partial Q_{21}}{\partial t} = \frac{4}{3} \cdot 0.5785 \cdot \sqrt{2 \cdot g \cdot \tan(\frac{\pi}{4}) \cdot (h'_x + 0.00009)}$$

Thus, the jacobian can be expressed as:

$$\begin{bmatrix}
K_{11} & K_{12} & K_{13} & 0 & 0 \\
K_{21} & K_{22} & 0 & 0 & 0 \\
0 & K_{32} & 0 & K_{44} & K_{45} \\
0 & 0 & 0 & K_{54} & K_{55}
\end{bmatrix} \begin{bmatrix}
K_x \\
K_{12} \\
0 \\
K_{23} \\
0 \\
K_{34} \\
0 \\
K_{35} \\
0 \\
0
\end{bmatrix}$$

where,

$$K_{11} = \frac{1}{A} \left( \frac{\partial Q_{12}}{\partial h_h} + \frac{\partial Q_{13}}{\partial h_h} \right), \quad K_{12} = \frac{1}{A} \left( \frac{\partial Q_{12}}{\partial h_x} + \frac{\partial Q_{13}}{\partial h_x} \right), \quad K_{13} = \frac{1}{A} \left( \frac{\partial Q_{12}}{\partial h_y} + \frac{\partial Q_{13}}{\partial h_y} \right)$$

$$K_{21} = \frac{1}{A} \left( \frac{\partial Q_{12}}{\partial h_h} + \frac{\partial Q_{13}}{\partial h_h} \right), \quad K_{22} = \frac{1}{A} \left( \frac{\partial Q_{12}}{\partial h_x} + \frac{\partial Q_{13}}{\partial h_x} \right), \quad K_{23} = \frac{1}{A} \left( \frac{\partial Q_{12}}{\partial h_y} + \frac{\partial Q_{13}}{\partial h_y} \right)$$

$$K_{31} = \frac{1}{A} \left( \frac{\partial Q_{12}}{\partial h_h} + \frac{\partial Q_{13}}{\partial h_h} \right), \quad K_{32} = \frac{1}{A} \left( \frac{\partial Q_{12}}{\partial h_x} + \frac{\partial Q_{13}}{\partial h_x} \right), \quad K_{33} = \frac{1}{A} \left( \frac{\partial Q_{12}}{\partial h_y} + \frac{\partial Q_{13}}{\partial h_y} \right)$$

$$K_{41} = \frac{1}{A} \left( \frac{\partial Q_{21}}{\partial h_h} + \frac{\partial Q_{21}}{\partial h_h} \right), \quad K_{42} = \frac{1}{A} \left( \frac{\partial Q_{21}}{\partial h_x} + \frac{\partial Q_{21}}{\partial h_x} \right), \quad K_{43} = \frac{1}{A} \left( \frac{\partial Q_{21}}{\partial h_y} + \frac{\partial Q_{21}}{\partial h_y} \right)$$

$$K_{51} = \frac{1}{A} \left( \frac{\partial Q_{21}}{\partial h_h} + \frac{\partial Q_{21}}{\partial h_h} \right), \quad K_{52} = \frac{1}{A} \left( \frac{\partial Q_{21}}{\partial h_x} + \frac{\partial Q_{21}}{\partial h_x} \right), \quad K_{53} = \frac{1}{A} \left( \frac{\partial Q_{21}}{\partial h_y} + \frac{\partial Q_{21}}{\partial h_y} \right)$$

$$K_x = \frac{1}{A}$$

And for $x' < 0.5$

$$K_{15} = \frac{1}{A} \left( \frac{\partial Q_{13}}{\partial h_h} \right), \quad K_{25} = \frac{1}{A} \left( \frac{\partial Q_{13}}{\partial h_x} \right), \quad K_{35} = \frac{1}{A} \left( \frac{\partial Q_{13}}{\partial h_y} \right)$$

or $x' > 0.5$

$$K_{15} = \frac{1}{A} \left( \frac{\partial Q_{13}}{\partial h_h} \right), \quad K_{25} = 0, \quad K_{35} = \frac{1}{A} \left( \frac{\partial Q_{13}}{\partial h_y} \right)$$

### 3.3 Multi-rate Discrete Model

Reference inputs are given at a sampling time period of $T = 3.3$ sec. However, this sampling rate is inappropriate for the control loop and Kalman filter, which has been selected to be at least $T_e = 0.1$ sec. In order to obtain reference inputs at a higher frequency, the ideas of (Armesto and Tornero 2003) have been implemented. The idea is to generate a continuous-time signal from the low-frequency discrete time reference inputs. Then, a discretization at faster sampling period of the continuous time signal is performed, where the ratio between these sampling periods is $N=3.3$. The main difference between the approach followed in (Armesto and Tornero, 2003) and the one use in this paper is that reference is complete known and can be off-line preprocessed. In this case, Splines (Cox, 1975) has been used to interpolate reference inputs.

In addition to this, the control sampling period $T_c$ has been limited to $T_c=0.1$ sec, because of the computational time of the algorithms. However, for an appropriate approximation of the discrete-time model of equation (1), a higher sampling period $T_s$ is internally required. Otherwise the model presents some numerical errors, particularly when heights are close to 0m. In our case, $T_e=0.02$ sec has been shown to be appropriate enough, thus $K=5$ is the ratio between the control loop and the internal flood simulator. This multirate control structure is shown in figure 7.

![Figure 7 Multi-rate control loop](image)

### 3.4 Simulation Results

The following parameters have been used for our simulations:

- $A_1=0.784$ m$^2$, $A_2=0.32$ m$^2$, $A_3=0.16$ m$^2$, $A_4=0.8$ m$^2$, $A_5=4$ m$^2$, $l_{12\text{Max}}=0.8$ m, $l_{13\text{Max}}=0.8$ m, $a_{21}=0.09$ m$^2$,
- $a_{45}=0.08$ m$^2$, $C_{DF} = C_{D2}, C_{D3}, C_{D45}=0.75$. Figure 8 shows a comparison of the reference flood flow depicted in figure 1 and the simulated flood flow, where it can be appreciated that the error is negligible.
4. RESERVOIR SIMULATOR DYNAMIC MODEL

Detailed description of the reservoir simulator is depicted in figure 11.

Figure 11 Reservoir simulator

Flood flow is the input of tank “Td”. The goal of this part is to reproduce the behaviour of the reservoir under the three situations as depicted in figure 1. Two sonar sensors are used to measure the height of the tank. Surplus water is evacuated through the Dam-gate. Next figure shows the representation of modelled tank in order to implement an appropriate control law.

Figure 12 Modelled tank

Equations to represent dynamic behaviour are:

\[ Q_f(t) - Q_d(t) = A_d \cdot \dot{h}_d(t) \]

\[ Q_d(t) = C_{DG} \cdot l_g \cdot \dot{h}_g(t) \cdot \sqrt{2 \cdot g \cdot h_d(t)} \]

Where \( Q_f(t) \) is the flood flow generated by the flood simulator, \( Q_d(t) \) is the volume of water evacuated for the Dam-Gate, \( A_d \) is tank area, \( C_{DG} \) is the Discharge coefficient of the output valve, \( h_g(t) \) and \( l_g \) are the height and the length of the Dam-Gate respectively.
4.1. CONTROL ARCHITECTURE FOR THE RESERVOIR SIMULATOR

Next figure shows control loop for the reservoir simulator.

![Figure 13 Multirate control loop](image)

In this case, flood flow “Qf” is introduced in this control loop as a perturbation of the reservoir simulator. Kalman observer is not necessary in this control structure because the measure is directly given by the sonar. It can be appreciated that the reference has been interfaced also at a higher frequency as commented section 3.3.

4.2 SIMULATION RESULTS

In this case, parameters used for the reservoir simulator are:

\[ A_r = 7.5 \text{ m}^2, \quad l_f = 0.4 \text{ m}, \quad C_{Df} = 0.75 \]

Next figure shows the simulation of the three behaviors of the reservoir depicted in the figure 1. Figure 14 also shows that the error is also neglectible.

![Figure 14 Simulated reservoir behaviour](image)

5. SCADA

In order to supervise and control the overall application, capable of dealing with system failures, a state machine based on GEMMA (ADEPA, 1992) has been implemented in a SCADA software interface. GEMMA schema is shown in figure 15.

![Figure 15 State machine](image)

When the state machine turns-on, “Initial state” is activated and the main page is shown in the SCADA application. (See Figure 16).

![Figure 16 SCADA](image)

When simulation references are selected, state machine moves to state “Make ready”. The user has three buttons in the left top of the page: “Play”, “Pause” and “Stop”. If the user pushes “Play”, state machine changes to the state “Simulation”. In this state, flood and reservoir simulator are run with the selected references. If the user pushes “Stop” button, both simulators are stopped, and, state machine changes to the state “Terminate”, and then, to the “Initial state”. If the user push the button “Pause”, the state is changed to the “Stop in the selected reference”, which maintains current references for the flood and reservoir simulator in order to study and measure the position and dimensions of the wave in the physical relief at its stationary behaviour.

When the process is simulating, two additional pages of the SCADA are enabled, reservoir page and flood page. In the flood page, real-time state of the flood simulator is presented. Flood reference selected in the main menu, is compared with current real-time flood flow measured by the sonar. Another plot shows, in real-time,
7. CONCLUSIONS & FUTURE WORKS

This paper describes an adopted solution to develop a physical reservoir and flood simulator in order to study the wave formed when the flood water collides with the reservoir water in the Mijares River (Valencia, Spain).

Numerical models have shown not to be a valid solution for this kind of problems and therefore, physical scaled model must be developed. For this purpose, simulators of the flood water and reservoir are required, which are modeled as a complex system of coupled tanks. Robustness and precision are also required for the application, since scale magnifies the errors.

Results show the accuracy of the proposed control system, where multi-rate techniques have been used in order to improve the performance.

A SCADA interface based on GEMMA state machine has been developed. This software is able to deal with system failures as well as to monitor and control the overall application.

8. ACKNOWLEDGEMENT

Authors wish to be grateful to A. Duran for the design of SCADA interface and V. Franch for the mechanical design of the reservoir gate.

9. REFERENCES


ADEPA. "GEMMA. "Guide d’Etude des Modes de Marches et Arrêts"). 1992

CEAM “Influence of Air Pollution in the Northern Region of the Valencian Community” Technical report. 1998


Y. Bar-Shalom, X. Rong Li, T. Kirubarajan. "Estimation with Applications to Tracking and Navigation". 2001

6. HARDWARE ARCHITECTURE

The hardware architecture is based on an industrial PC used for monitoring and controlling the simulated model. Figure 17 shows hardware architecture with the following sensors and actuators:

- Two stepper motors controlling the position of the dam gate and the flood gate.
- Two stepper drivers connected with the PC by RS-232 Net
- Two sonars measuring the height of tanks of the flood and reservoir simulator. This sensors are connected to the PC with a CAN bus.
- Four end switches to detect open-close gates

![Figure 17 Hardware architecture](image_url)
SIMULATION BASED TRAINING AND ASSISTANT SYSTEMS
FOR BIOGAS FACILITIES

Alexei Lisounkin, Alexander Sabov, Gerhard Schreck, Jörg Krüger
Fraunhofer Institute for Production Systems and Design Technology
Pascalstrasse 8-9, D-10587, Berlin, Germany
E-mail: { alexei.lisounkin | alexander.sabov | gerhard.schreck | joerg.krueger } @ipk.fraunhofer.de

KEYWORDS
Simulation in energy and power systems, Web service architecture, assistance and training, facility and process modeling, client-server communication

ABSTRACT

This paper is devoted to software engineering and modelling aspects for Web service-oriented process supervision assistant and training systems. The development focuses on the system architecture, modelling techniques and process simulation integration. An assistance and training system for biogas facilities constitutes the application field of the results.

INTRODUCTION

Biogas facilities are chemical reactors where microbes produce hydrogen under anaerobic conditions and methane and carbon monoxide by means of biomass gasification. Even if the process is rather slow, under certain conditions it becomes unstable. In the case of a malfunction and the resulting failure, the loss of profits could be remarkable. As the result, the economic efficiency of the facility can be negative. Obviously, the facility operator needs good process knowledge.

The use of simulation-based training and assistant systems on site as well as by remote access will increase the efficiency of facility operations by means of increasing the qualifications of the operating staff and optimizing the involved technology.

In general, biogas facilities are operated by small enterprises and/or farmer associations. The facilities differ substantially from one to another, and the raw materials may vary as well. The facility operation assistance and training tasks require high level of process knowledge, an advanced reaction modelling, archives of valid operation trends, and an adequate IT infrastructure. It may be considerably expensive to develop the necessary facility operation assistance and training tools.

The use of Web technologies offers the possibility of developing and implementing such tools on a Web-based business: A single advanced assistance and training system with a powerful simulation core will serve several biogas facilities and several clients as well (Schreck 2002).

This paper considers a general architecture for such a system, general modelling principles, relevant technological issues with respect to IT infrastructure and simulation, as well as some partial aspects, e.g. a client-server interaction mechanism and the integration of the simulation core into the Web service.

SYSTEM ACTORS AND USE CASES

The first step in developing a system includes defining the main actors and external components which interact with the system. In the case of an assistance and training system for a biogas facility, the main use case involves the interaction between a facility operator and a facility model. User access to other roles – technologist, facility engineer, planning administrator – is also simultaneously necessary. In the case of training tasks, trainee and trainer roles need also be foreseen.

In view of the demands on the availability and economic efficiency of such an assistance and training system, corresponding use cases have been considered. Considering the system availability and maintenance requirements in context of communication possibilities, a Web-services oriented architecture will provide an efficient solution for the assistance and training system (Page 2000).

![Figure 1. Use cases](Image)
Reflecting the considerations above, the use case and system resources diagram can be represented as shown in Figure 1.

Thus, the involved actors are:
- A facility operator, trainee,
- system administrator,
- an optional training instructor,
- facility and maintenance engineer,
- technologist.

The main system use cases are:
- system and training management,
- model configuration,
- execution of training sessions.

**SYSTEM ARCHITECTURE**

The main focus of the design of the assistance and training environment is the costs for the hardware, software, system development, setup and maintenance. The proposed architecture allows for the development of low cost solutions by (c.f. Lisounkin 2004):

- using thin clients which require minimal installation and maintenance support for the facility operator,
- using global and readily accessible communication methods such as the Internet,
- using standard Web browsers for the front end, as available with any low-cost standard PC,
- using a centralized simulator that serves multiple clients,
- integrating expert tools into the assistance and training environment.

Figure 2 presents the general system architecture for the approach.

![General system architecture](image)

Figure 2. General system architecture

The components are separated according to the Model-View-Controller (MVC) concept. While the graphical user interface resides on the operator machine, the business logic and data is located on a central server. This allows the separated components to be exchanged. In our approach we mainly use three front ends for the **Training GUI** which all work with the same business logic:

- a Java-based GUI for the training instructor,
- a Java-based GUI for the trainee,
- a HTML-based GUI for administration and maintenance.

The GUI is completely restricted to user interaction and visualization so the business logic can reside on the server and serve all front ends at once. Contact to the server is achieved through the **Training Client**, which acts as a communication broker using the Apache Axis library for Java. The main tasks of the training client are to serialize the data and execute the remote procedure call over the Internet. Process variables are sent to and received from the simulator and mapped to the front end GUI components.

On the server side, client requests are handled by the **Web Service** component which is the interface to the business logic. At this point we specifically make use of the Web service philosophy, separating tasks into independent modules and defining the interfaces from the functional point of view rather than according to application-specific details. This way, the Web service modules can be grouped together for different purposes and do not depend on only one application. One main Web service is the **Training Service**, which offers general training-specific functions such as adding clients to a training session and managing the training progress. A special script mechanism allows running predefined scenarios for hazards and abnormal situations.

The **Simulator** can be separated into two main components, the simulator control and the computation core. The main tasks of the **Simulator Control** are to execute the start/stop commands and to manage simulator instances in order to be able to serve several clients in parallel. Since most processes in the biogas facility elapse very slowly, a time-referenced simulation must be implemented, allowing the process behaviour to speed up. Further tasks of the simulator control are the management of model instances, their updates and simulation trends.

The **Computation Core** contains the solver for the differential equations defined by the process model. Due to the Web service philosophy, multiple solvers can be used together with their corresponding models. Considering the requirements of a low-cost system, the integration of expert tools on the market is of high importance. In our approach we integrated the Matlab / Simulink software into the training environment by creating a special Web service adapter for the Matlab engine (Figure 3).
The **Training Management Database** contains all data necessary for training. This includes:

- biogas process models with hazard scenarios,
- user profiles and qualifications,
- access to historical training sessions,
- exercise schedules with deadlines for successful training.

The **Biogas Process Model** is based on an advanced modeling language and contains the following data:

- facility components and associated process parameters and equations,
- the topology of the facility,
- process evolution profiles for several real facilities,
- initial values of the process parameters.

Figure 3 presents the system architecture for the client machine and the simulation server. Applying the thin client concept, the client machine mainly requires a web browser for accessing the simulation server. For complex user interfaces, Java applets are used as well.

![System architecture of the Simulation Server](image)

Incoming simulation requests are received by the simulation Web service component from the client interface. The requests are then forwarded to the Matlab engine manager. Since the Matlab software is only one possible simulation core, simulator requests are described in a generic way and must be translated into Matlab specific commands by the Matlab engine manager.

**CLIENT-SERVER COMMUNICATION**

The approach of a general browser-based communication format: between the operator and the training service requires client/server technologies that are independent of the underlying hardware and software platform. Here we consider a solution using the SOAP/Web service technology. SOAP is a standard protocol intended for exchanging structured information in a decentralized, distributed environment (http://www.w3.org/TR/soap/) and can be seamlessly integrated into HTTP requests from a web browser. The browser contacts a specific web service and makes a remote procedure call, sending the input parameters and delivering the output parameters back to the client. The interface for the available procedures of the Web service is defined using the XML-based specification WSDL (Web Services Description Language (http://www.w3.org/TR/wsd1)). The client and server side can be developed rapidly for common programming languages using code generation tools.

Since current Web service standards do not define transaction states a special state machine has been implemented which also considers simulator-specific transitions. Figure 4 presents a simplified version of the state machine.

![State machine for simulation control](image)

When a new client connects to the system, the client calls the **New** command and sends the initial biogas process model to the simulator. The simulator resides in the **Init** state until all initialization tasks have been finished (model parsing, interpretation, computation setup, etc).

The computation of the process model is divided into two parts. The first part consists of the **ReadyListening/Listening** loop and is used for pre-calculations:

- setting initial values for the process parameters,
- setting initial values for the approximation methods,
• calculating intermediate results for speed enhancements.

The second part, the *ReadyComputing/Computing* loop, is responsible for computing the main process. Here the differential equations which depend on the facility components and the topology are solved. The main tasks are:

• solving the differential equations,
• synchronising the computation task between all solvers in the *ReadyComputing/Error* state.

**SIMULATION MODEL AND PROCEDURE**

A simulator for the biomass fermentation process constitutes the core of the assistance and training system. There are currently different techniques to describe this process (see Friedmann 1993, Abbasi, 2005), and the process model is still under development.

The process model shown in Figure 5 is divided into three main phases. The process starts with the filling of the raw material into a fermenter. The raw material typically consists of bio waste such as liquid manure, sewage sludge, fats, or plants.

![Diagram of fermentation process](Image)

**Figure 5. Phase model for methane fermentation based on Friedmann 1993**

In the first phase, the high-structured molecular compounds are split into their basic constituents. A fermentation process then converts the product into different acids and other compounds. The entire process is accomplished by hydrolytic and fermentation bacteria. Since the process modelling is currently not completely finished, a data oriented approximation model based on measurements from a real facility was used for the hydrolysis phase.

The second phase focuses the formation of acetic acid. Acetogenic bacteria convert the low-structured molecular organic acids and alcohols into acetic acid, hydrogen and carbon dioxide. In the first and second phase, a low concentration of hydrogen is of high importance for optimal process activity.

In the final phase, the biogas consisting of methane, hydrogen sulphide and carbon dioxide is formed by methanogenic bacteria. The biogas is mostly used for decentralized block heat and power plants, though a part of the heat created is needed to maintain the fermenter temperature.

The simulation is based on dynamic model that describe the reactor behaviour depending on available inputs. The acetic acid forming and the methane forming phases have been modelled by means of ODE's. Moreover, the phenomena of methane diffusion into bubbles and bubbles dynamic in the reactor have also been considered by the model.

Regarding the assistance and training tasks, the simulation model and procedure must to fulfil the following requirements (NAMUR 1995):

• The model must possess a specification of the state variables used by the simulation procedure which provides a unique model evolution trend.
• The simulation procedure must allow for the intermediate values of state variables to be saved and accepted as the starting point of a new simulation run.
• The simulation procedure must be able to provide an accelerated (with respect to real time) evolution of a facility trend with defined acceleration coefficients.
• The simulation procedure must provide a history of the facility evolution for its analysis.

To provide for the simulation service over the Internet, the following conditions must be met:

• The representation of the system dynamics at the client site must be provided monotonously and smoothly with respect to time.
• Soft real time for the client-server communication must be realized.
• The volume of the client-server information exchange must be minimal.

**APPLICATION**

The concepts and techniques presented in this paper were applied to the design and development of an assistance and training system for biogas production facilities.

User access to the system was via a standard Web browser (e.g. Microsoft Internet Explorer, Mozilla Firefox, etc.). An
example of a facility operation mask is given in Figure 6.

For the visualization of dynamic processes – e.g., process measurements and simulation state evolution – communication between the client and server must be continuous. For a graphical visualization of simulator inputs and process trends, a Java applet was developed. A visualization mask for process trend graphics is given in Figure 7. In the example, the bacteria concentration trend in the reactor was simulated for a period of 150 hours. Then, the reactor temperature was increased from 55°C to 80°C. The heat lets the bacteria die off in this period, as shown in the figure. The operator reacted to the unstable state at time 300 and set the reactor temperature back to 55°C. The operator acted too late, as the bacteria did not regenerate.

CONCLUSION

The research and development presented here provides an applicable concept for the implementation of advanced Web service oriented assistance and training systems on a low-cost basis. The application domain for such systems is any technological area where small and medium-sized enterprises play a significant role. Usually, the variety of application contexts and the specific requirements for technology dimensioning require the involvement of various specialists in the production process. The Web-based simulation system can be a communication platform for technology analysis and implementation.

ACKNOWLEDGEMENT

The presented architecture described was developed within the framework of the R&D project "IT-basierte Systeme und Verfahren für die sichere Produktion". This project was supported by public state funds and the European Fund for Regional Development (EFRE) within the scope of the technology initiative “Safety with IT” by the Senate of Berlin. We would additionally like to thank our industrial partner AUCOTeam GmbH, Berlin, Germany, for their extensive cooperation.

REFERENCES

DISTANCE LEARNING: TESTING AN INTERACTIVE MATHEMATICAL COURSE

Tiziana Durante and Annunziata Cascone
Department of Information Engineering and Applied Mathematics
University of Salerno,
via Ponte don Melillo, 4084 Fisciano (SA), Italy
E-mail: [durante, cascone]@diima.unisa.it

Rosaria Menna
High School I.P.C. Miano
Viale della Resistenza, 167 loc. Miano
80100 Napoli, Italy

KEYWORDS
Computer Based Education, Active Learning, Interactive tools, Queueing Theory.

ABSTRACT

The evaluation of a learning environment should be based on modern learning theories. Computer based activities have to be integrated into traditional ones in order to have a positive impact on learning and to produce true changes in the teaching-learning model. Computer aided learning applications are able to offer advanced students the opportunity to improve their skills and to keep their motivations. This paper deals with evaluation of hypermedia-based learning environments mainly from the point of view of the learner or student. Its aim is to present an experimentation of an interactive course implemented on Internet thanks to new technologies of e-learning. This advanced e-course, whose contents are about Queueing Theory, has been tested on a class of students of Electronic Engineering Faculty by means of the IWT platform.

INTRODUCTION

In the last twenty years Internet became an handy tool in the teaching-learning process in Universities as well as in Industries. Interactive courses are developed with the aim to be used by an increasing number of learners in different contexts. They help students to reconstruct and enrich existing knowledge, giving them chances to explore, interpret phenomena, collaborate, ask questions and share ideas. Universities are playing more and more leading role in the supply of contents, services and solutions for e-learning, while it isn’t the same for the demand. In fact until now e-learning only consisted of some individual initiatives. Some multimedial didactical material was only arranged by some enterprising teacher to support his students training. So the courses on-line, today Universities offer, are the minority and they only have experimental features.

In this background, the IWT (Intelligent Web Teacher) Platform represents an innovative instrument that allow to lead e-learning on a larger scale providing the opportunity to a greater number of teachers to realize interactive sources for distance learning. This platform, in fact, doesn’t need special skills because of its simple usability.

This paper presents testing phases and results of interactive classes on Queueing Theory taken for students of Engineering Faculty at Salerno University.

It is set as follows. In section 2 we describe the common features and some specific tools of IWT Platform. Section 3 points up, from a more practical point of view, the applications and procedures of this conduct test. In section IV we talk about evaluation and feedback, while conclusions are drawn in section V.

DESCRIPTION AND GENERAL FUNCTIONALITIES OF IWT PLATFORM

Platform IWT is an innovative interactive instrument for distance learning. It can be tailored, according to the didactic users’ preferences, to deliver advanced “intelligent” courses in which the strategies and the didactic path are dynamically personalized to comply with the learning time proper of any single individual.

In IWT platform is possible to create the so called “Intelligent Courses”, that is didactic courses dynamically created in according to the previous knowledges, the preferences and the attitudes of learning of each student who take up the on-line lectures.

To create this type of course it is necessary to explain the domain of knowledge with an ontology, i.e. a graph structure consisting of various atomic concepts related through two types of different relations as “Has Part” and “Is Required
By” ones that allow to link all the learning concepts in a conceptual map.

The contents of the course, managed by students, are introduced by LO (Learning Object). To design a course, the teacher has to fix didactic aims and to define times and modalities of assessment with the so-called milestones thanks to that he can appraise the ability of his students.

Then the platform itself locates, in the global ontology of the knowledge domain, the section of the graph involving subjects of the defined course, so automatically making the ontology of that course. Student’s profile (his attitudes, knowledge) is realized at the beginning and revised during its development so that the course can be dynamic proposing to the learner those LO related to the not understood concepts.

**TESTING IWT PLATFORM ON MATHEMATICS V COURSE**

In this last year, during lectures of Mathematics V about Queueing Theory held at the Electronic Engineering Faculty of Salerno University we tested this innovative platform IWT on students class attending this course.

Together with theoretical lectures in the classroom, students could access to a parallel interactive course published on Internet where they could find an Intelligent Course on Mathematics V subjects. This course consists of theoretical lessons that can be glanced through on line in a text form. All these concepts are shown in an order chosen by ontology according to personalized path created as a result of an opening test that each student has to attempt. Consulting resources the user has to answer periodically to a series of tests by means of them the platform evaluate student’s knowledge and, every time, it will remake the personalized path in agreement with obtained results. In particular the platform proposes again concepts that it considers not well acquired by the student-user.

Moreover IWT platform offers a lot of other complementary services that allow a direct virtual interaction teacher-student and student-student. As an example, student can ask some questions directly to his teacher writing a message box. Then he can communicate with other students by means of a chat and a forum where he can compare himself with them on the course contents and every other thing on which he want to add a thread.

Others platform functionalities consist of the chance to take part in videoconference and the option to make on-line some homework assigned by teacher who can also correct them in real time so reducing by far times and distance. The teacher also can take some advantages in control of his class by checking reports of their performances. He can see progresses of each student monitoring for everyone acquired concepts, passed tests, solved problems and asked questions. In this way, both student and teacher are active protagonists of the course.

Now we give an example of a situation that occurs to a student who approaches to this interactive course. First of all, to enter he has to register himself at the platform.

After registration, he can enter writing his login and password. So he’ll visualize his home page with the possibility to approach to all the platform functionalities. For example, in the section “Courses” he can find all the published courses and he can send a request for booking to the selected course.

When both the administrator and the teacher give the authorization, finally the student can enter into the course and begin studying according his personal study path.

Each student who enters into Mathematics V Intelligent Course can do the following things:

- Study theoretical resources reading the didactical units on line (see Fig. 1);
- Evaluate himself by solving some interactive tests consisting of multiple choice questions (see Fig. 2);
- Solve some exercises guided step by step by the platform itself (see Fig. 3);
- Interact with his teacher directly writing in a message box;
- Have a chat with other students by means of a “Chat” section and begin a talk or leave a message in a “Forum” section;
- Download some added didactical resources putted at students disposal by teacher in “Support Material” section;
- Do virtual homework on-line.
EVALUATION AND FEEDBACK

The comments we report below are the result of students feedback collected, at the end of the course, during a testing talk when a liking feedback form (see the Appendix) was submitted to each student. In general students appreciated the interactive course published on IWT platform, showing great interest. Almost everybody are fascinated and spurred by the possibility of interacting on line putting to the test one’s own knowledge. Feedback from students suggests that the use of the platform in the teaching-learning process has been a benefit for many of them. In fact many students consider it as an important aid to improve their training and to develop self-confidence. Most of them think that it has been more than sufficient in getting better their comprehension not only in view of final examinations, but also as a personal enrichment in terms of both an electronic approach and subject involvement. During the debate it also emerged that having to deal with interactive tests and exercises can increase the consciousness of knowledge because by making them, they take upon themselves some responsibility for their own learning. Comparing students who attended these lectures with those who, during previous year, only attended the traditional ones setting in classroom, on the basis of results obtained in the final examinations there is a significant increase in successful candidates (->9%) and a slight rise in the average marks (->3%). This result is sure an extremely good starting point, but we need to go on with testing to be able to confirm or not these positive results. Considering the achieved outcomes, we cannot help trusting! Most students who volunteered for this testing thought computer learning rather enjoyable because it encouraged them so much that they felt very fulfilled. Nearly every student agrees that they did not consider these computer based lectures as “ordinary boring” ones, but as a pleasant, interesting and unusual opportunity for studying, so a useful tool from an educational point of view!

We are very excited about these first results even though we have to point out few disadvantages. As an example we have to control the risk that some students could depend on the computer response specially if aided step-by-step exercises in comparison with those ones made by themselves contribute to increase their lack of security instead of gaining confidence. This aspect might be worrying since students could not be able to become self-sufficient, but luckily this effect is quite trivial ever since, till now, it was only an uncommon event.

CONCLUSION

Thanks to new technology, we have the opportunity to enhance the teaching-learning process, improve learner-control, increase motivation and establish connection to the real world. The understanding of how Computer Based Education (CBE) can be designed and implemented is the focus on which we have to address our attention as well as the pedagogical importance of the use of CBE in the learning process.

In any case, the human and contextual factors influence the success or failure of technological tools more than the developed applications. The research on technology’s effectiveness on educational uses is widespread and, in some cases, disappointing in quality, but it is typical since the subject of study is a moving target. The social alienation is a key problem in distance education. It arises from the removal of students from their colleagues and the instructor and so there are strong social motivations for Internet use in the distance education classroom as our experiment pointed out.

We hope that the framework for teaching prospects, presented in this paper, may provide a shared structure for monitoring student progress and planning research.

REFERENCES


Berrynan S. E.: Learning for the workplace, Review of research in Education, 19, 343-401.


http://www.momanet.it.

BIOGRAPHY

TIZIANA DURANTE was born in Rome, Italy and obtained PhD Degrees in Mathematics in 1997. She is a researcher at the Department of Information Engineering and Applied Mathematics in University of Salerno. Her research interests include: Computer Aided Learning, Hybrid Systems, Homogenization Problems, Spatial Behaviour for Dynamic Problems.

ANNUNZIATA CASCONE was born in Castellammare di Stabia, Italy. She graduated cum laude in Mathematics in 2004 She is a actually a PhD student in Mathematics at the University of Salerno. Her scientific interests are about Computer Aided Learning, Queueing Theory and fluid-dynamic models for traffic flows on road and telecommunication networks.

ROSARIA MENNA was born in Castellammare di Stabia, Italy. She graduated cum laude in Mathematics in 2004. She is a actually a PhD student in Mathematics at the University of Salerno. Her scientific interests are about Computer Aided Learning, Queueing Theory and fluid-dynamic models for traffic flows on road and telecommunication networks.
APPENDIX

Feedback Form for Students

1. Was the theoretical lecture helpful for exercises resolution?
   - Very Helpful
   - Helpful
   - Somewhat Helpful
   - Not Helpful

2. Are the step-by-step exercises easy to follow and good for self-practicing?
   - Very Easy
   - Easy
   - Somewhat Easy
   - Not Easy
   - Good
   - Sufficient
   - Somewhat Sufficient
   - Insufficient

3. What do you think of time devoted to platform testing?
   - More than sufficient
   - Sufficient
   - Somewhat Sufficient
   - Insufficient

4. Do you think that computer-training is enough for your exam?
   - Enough
   - Not Enough

5. Did you practice only this platform exercises or other sources as exercises-textbooks, too?
   - Only This Package
   - Also Other Sources

6. What do you think of this platform?
   - Very Useful
   - Useful
   - Somewhat Useful
   - Not Useful

7. Do you think that teaching would be the same without IWT course?
   - Yes
   - No

8. Do you think that would be suitable to repeat this experience next years?
   - Yes
   - No
   - I Don’t Know

9. Did you enjoy learning by doing with this interactive course?
   - Yes
   - No
   - Not enough

➢ Please write your additional comments and advices here
COMPLEX
SYSTEMS
SIMULATION
DISCOVERING RELATIONSHIPS THAT ARE NOT FUNCTIONS

William Conley
Professor of Business Administration (Statistics)
University of Wisconsin at Green Bay
Green Bay, Wisconsin 54311-7001, U.S.A.
email: conleyw@uwgb.edu

KEYWORDS
CTSP statistic, correlation, spherical relationships, business example

ABSTRACT
Presented here is the new CTSP correlation coefficient applied to a three variable and a nine variable data set, where strong relations exist, but they are not in the form of a function. This can happen in practical science, business and economics problems. Therefore, it is important to be able to detect these subtle, perhaps, “hidden” relationships when they exist. The three variable problem will help to illustrate the ideas and the technique involved here, while the second example will be a practical management and operations research application. The simulation based multi stage Monte Carlo technique will be used to solve the CTSP (correlation with the traveling salesman problem) examples presented here.

INTRODUCTION
It could be argued that the most difficult recurring problem in applied statistics is the analysis of large sets of multivariate data to see if the variables that gave rise to the data are related in some fashion or not. The Pearson r correlation coefficient can measure a linear relationship between variables. Its larger generalization, big R (Anderson 2003), can measure linear relationships between three or more variables. The new computer simulation based CTSP family of correlation coefficients have been used to discover linear and nonlinear functional relationships. A cubic equation relationship was identified in (Conley 2005) for example and other relationships in (Conley 2003).

However, it should be noted here that the CTSP approach for measuring correlation is completely general and also should work on data sets that are correlated in a mathematical relation sense, but not in functional sense. Many strong and useful relations are not functions. Let us test this claim by looking at a three variable data set and a nine variable profit maximization business problem.

A THREE VARIABLE RELATION
The following n=119 readings of X₁, X₂ and X₃ were taken by a researcher who believes that the three variables are highly correlated. However, big R (the multivariate linear correlation coefficient) is near zero for the data. Also, the data seems to defy all attempts at fitting a function to it (with little error). However, a relation is possible, where g (X₁, X₂, X₃) can take more than one value for given X₁, X₂, X₃. A function requires a unique value f (X₁, X₂, X₃) for each triple. However, a relation (such as an ellipse) does not.

Table 1: Three Variable Data

<table>
<thead>
<tr>
<th></th>
<th>X₁</th>
<th>X₂</th>
<th>X₃</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>85</td>
<td>37</td>
<td>83</td>
</tr>
<tr>
<td>2</td>
<td>55</td>
<td>2</td>
<td>63</td>
</tr>
<tr>
<td>3</td>
<td>41</td>
<td>43</td>
<td>99</td>
</tr>
<tr>
<td>4</td>
<td>36</td>
<td>95</td>
<td>67</td>
</tr>
<tr>
<td>5</td>
<td>59</td>
<td>84</td>
<td>86</td>
</tr>
<tr>
<td>6</td>
<td>23</td>
<td>73</td>
<td>85</td>
</tr>
<tr>
<td>7</td>
<td>56</td>
<td>1</td>
<td>58</td>
</tr>
<tr>
<td>8</td>
<td>99</td>
<td>58</td>
<td>56</td>
</tr>
<tr>
<td>9</td>
<td>18</td>
<td>29</td>
<td>82</td>
</tr>
<tr>
<td>10</td>
<td>24</td>
<td>77</td>
<td>83</td>
</tr>
<tr>
<td>11</td>
<td>28</td>
<td>77</td>
<td>86</td>
</tr>
<tr>
<td>12</td>
<td>54</td>
<td>89</td>
<td>81</td>
</tr>
<tr>
<td>13</td>
<td>22</td>
<td>88</td>
<td>67</td>
</tr>
<tr>
<td>14</td>
<td>13</td>
<td>76</td>
<td>71</td>
</tr>
<tr>
<td>15</td>
<td>43</td>
<td>87</td>
<td>83</td>
</tr>
<tr>
<td>16</td>
<td>30</td>
<td>11</td>
<td>74</td>
</tr>
<tr>
<td>17</td>
<td>37</td>
<td>70</td>
<td>94</td>
</tr>
<tr>
<td>18</td>
<td>77</td>
<td>89</td>
<td>66</td>
</tr>
<tr>
<td>19</td>
<td>32</td>
<td>25</td>
<td>89</td>
</tr>
<tr>
<td>20</td>
<td>99</td>
<td>48</td>
<td>60</td>
</tr>
<tr>
<td>21</td>
<td>27</td>
<td>48</td>
<td>94</td>
</tr>
<tr>
<td>22</td>
<td>70</td>
<td>13</td>
<td>77</td>
</tr>
<tr>
<td>23</td>
<td>78</td>
<td>65</td>
<td>89</td>
</tr>
<tr>
<td>24</td>
<td>61</td>
<td>20</td>
<td>89</td>
</tr>
<tr>
<td>25</td>
<td>52</td>
<td>12</td>
<td>82</td>
</tr>
<tr>
<td>26</td>
<td>44</td>
<td>24</td>
<td>92</td>
</tr>
<tr>
<td>27</td>
<td>69</td>
<td>36</td>
<td>94</td>
</tr>
<tr>
<td>28</td>
<td>64</td>
<td>63</td>
<td>96</td>
</tr>
<tr>
<td>29</td>
<td>80</td>
<td>11</td>
<td>59</td>
</tr>
<tr>
<td>30</td>
<td>85</td>
<td>69</td>
<td>80</td>
</tr>
<tr>
<td>31</td>
<td>43</td>
<td>53</td>
<td>99</td>
</tr>
<tr>
<td>32</td>
<td>71</td>
<td>69</td>
<td>91</td>
</tr>
<tr>
<td>33</td>
<td>49</td>
<td>49</td>
<td>100</td>
</tr>
<tr>
<td>34</td>
<td>34</td>
<td>68</td>
<td>93</td>
</tr>
<tr>
<td>35</td>
<td>47</td>
<td>56</td>
<td>99</td>
</tr>
<tr>
<td>36</td>
<td>73</td>
<td>65</td>
<td>92</td>
</tr>
</tbody>
</table>
Thereore, the data is read into the multi stage Monte Carlo optimization (MSMCO) traveling salesman problem (TSP) algorithm (Conley 2000) adjusted for three variables. A shortest route of total distance $A=1312.760$ is calculated for connecting the points in a complete tour. This is the numerator of CTSP (the general purpose correlation coefficient). The route is to go from point 91 in a complete tour through all 119 points and return to point 91 in the following fashion. (The Pythagorean theorem was used repeatedly for the distance calculations.)

| 37 | 25 | 22 | 83 | 97 | 31 | 45 | 4 |
| 38 | 97 | 53 | 67 | 98 | 27 | 93 | 39 |
| 39 | 53 | 99 | 60 | 99 | 90 | 35 | 24 |
| 40 | 25 | 50 | 93 | 100 | 79 | 89 | 38 |
| 41 | 62 | 78 | 90 | 101 | 39 | 22 | 10 |
| 42 | 95 | 52 | 72 | 102 | 17 | 46 | 12 |
| 43 | 20 | 19 | 76 | 103 | 86 | 70 | 21 |
| 44 | 64 | 23 | 90 | 104 | 95 | 54 | 28 |
| 45 | 44 | 59 | 99 | 105 | 24 | 69 | 11 |
| 46 | 26 | 10 | 68 | 106 | 23 | 64 | 10 |
| 47 | 24 | 29 | 87 | 107 | 46 | 71 | 4 |
| 48 | 73 | 41 | 94 | 108 | 77 | 15 | 27 |
| 49 | 9 | 63 | 76 | 109 | 12 | 60 | 19 |
| 50 | 40 | 14 | 83 | 110 | 72 | 18 | 18 |
| 51 | 77 | 84 | 75 | 111 | 43 | 43 | 1 |
| 52 | 86 | 39 | 83 | 112 | 46 | 31 | 3 |
| 53 | 83 | 27 | 80 | 113 | 52 | 66 | 2 |
| 54 | 18 | 23 | 77 | 114 | 70 | 56 | 4 |
| 55 | 73 | 12 | 73 | 115 | 26 | 86 | 24 |
| 56 | 62 | 87 | 81 | 116 | 77 | 23 | 18 |
| 57 | 38 | 43 | 98 | 117 | 12 | 55 | 18 |
| 58 | 77 | 19 | 79 | 118 | 34 | 54 | 3 |
| 59 | 19 | 88 | 60 | 119 | 43 | 91 | 22 |
| 60 | 90 | 59 | 79 | 61 | 74 | 56 | 6 |
| 62 | 40 | 54 | 1 | 63 | 51 | 1 | 40 |
| 64 | 57 | 73 | 6 | 65 | 47 | 15 | 14 |
| 66 | 72 | 71 | 10 | 67 | 81 | 31 | 15 |
| 68 | 15 | 62 | 16 | 69 | 90 | 29 | 28 |
| 70 | 14 | 37 | 17 | 71 | 32 | 72 | 8 |
| 72 | 13 | 29 | 23 | 73 | 44 | 12 | 18 |
| 74 | 55 | 79 | 9 | 75 | 77 | 85 | 26 |
| 76 | 61 | 22 | 10 | 77 | 36 | 69 | 5 |
| 78 | 34 | 18 | 15 | 79 | 36 | 76 | 9 |
| 80 | 65 | 36 | 4 | 81 | 79 | 26 | 17 |
| 82 | 55 | 88 | 17 | 83 | 77 | 36 | 10 |
| 84 | 60 | 53 | 1 | 85 | 37 | 33 | 4 |
| 86 | 34 | 74 | 9 | 87 | 59 | 44 | 1 |
| 88 | 6 | 43 | 27 | 89 | 88 | 73 | 27 |
| 90 | 75 | 31 | 11 | 91 | 30 | 80 | 15 |
| 92 | 30 | 70 | 8 | 93 | 43 | 85 | 15 |
| 94 | 36 | 11 | 22 | 95 | 28 | 82 | 18 |
| 96 | 30 | 81 | 16 | 97 | 96 | 95 | 115 |

Table 2: The Route

| 91 | 96 | 95 | 115 | 98 | 59 | 13 | 4 | 39 | 15 |
| 12 | 56 | 5 | 41 | 51 | 18 | 100 | 75 | 89 | 103 |
| 66 | 61 | 114 | 84 | 87 | 80 | 83 | 90 | 67 | 81 |
| 116 | 110 | 108 | 69 | 99 | 104 | 8 | 20 | 38 | 42 |
| 60 | 30 | 23 | 36 | 32 | 28 | 33 | 31 | 35 | 45 |
| 17 | 34 | 11 | 10 | 6 | 14 | 49 | 40 | 21 | 57 |
| 3 | 26 | 19 | 37 | 47 | 9 | 54 | 43 | 46 | 16 |
| 50 | 25 | 24 | 44 | 27 | 48 | 52 | 1 | 53 | 58 |
| 22 | 55 | 29 | 2 | 7 | 63 | 76 | 65 | 73 | 94 |
| 78 | 101 | 85 | 112 | 111 | 62 | 118 | 97 | 102 | 70 |
| 72 | 88 | 117 | 109 | 68 | 106 | 105 | 92 | 71 | 86 |
| 79 | 77 | 107 | 113 | 64 | 74 | 82 | 93 | 119 | 91 |

Then five sets of $n=119$ triples in the same 0 to 100 range as the real data were produced using a random number generator. The MSMCO TSP algorithm (adjusted for three dimensions) found shortest complete tour routes of $2074.251, 1952.883, 2070.155, 2079.568$ and $1953.377$. Thereore, $CTSP=A/B=1312.760/2070.155=.6341$ using the median of the random data shortest route values for B.
Also, the $5 \times 4 = 20$ A/B quotients using the random data values (under the hypothesis of no correlation between $X_1$, $X_2$, and $X_3$) are all greater than or equal to 1952.883/2079.568=0.9391. Therefore, because CTSP=A/B=0.6341 using our real data, is much less than .9391, there is a strong relationship between $X_1$, $X_2$, and $X_3$.

It turns out that a pretty good fit for the data is equation (1)

$$(X_1-50)^2 + (X_2-50)^2 + (X_3-50)^2 = 2500 \quad (1)$$

which is a sphere (a relation but not a function). Therefore, the CTSP statistics seem to be able to detect any type of correlation between variables, be it functions of the linear or nonlinear type or nonfunctional relations.

Let us look at a management decision making problem with a few more variables.

**A NINE VARIABLE PROBLEM**

A subsidiary of a large company offers eight products to customers that are at least partial substitutes for each other. Therefore, the manager wants to set the eight prices for the products to maximize profit. The following data represents the eight prices charged for products one through eight ($X_1$, $X_2$, $X_3$, $X_4$, $X_5$, $X_6$, $X_7$, $X_8$) where the number is the percentage over basic cost to produce product $i$ that the price is set at. An example is $X_1=38$ means a 38% markup over basic cost to produce product one. Also, $X_9$ is the weekly corresponding profit (for the last 79 weeks of data) expressed in thousands of dollars.

<table>
<thead>
<tr>
<th>$X_1$</th>
<th>$X_2$</th>
<th>$X_3$</th>
<th>$X_4$</th>
<th>$X_5$</th>
<th>$X_6$</th>
<th>$X_7$</th>
<th>$X_8$</th>
<th>$X_9$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>38</td>
<td>48</td>
<td>47</td>
<td>61</td>
<td>48</td>
<td>53</td>
<td>96</td>
<td>47</td>
</tr>
<tr>
<td>2</td>
<td>42</td>
<td>52</td>
<td>32</td>
<td>43</td>
<td>73</td>
<td>44</td>
<td>28</td>
<td>40</td>
</tr>
<tr>
<td>3</td>
<td>72</td>
<td>42</td>
<td>20</td>
<td>58</td>
<td>59</td>
<td>42</td>
<td>40</td>
<td>30</td>
</tr>
<tr>
<td>4</td>
<td>73</td>
<td>70</td>
<td>76</td>
<td>34</td>
<td>43</td>
<td>58</td>
<td>64</td>
<td>34</td>
</tr>
<tr>
<td>5</td>
<td>50</td>
<td>50</td>
<td>50</td>
<td>50</td>
<td>50</td>
<td>50</td>
<td>50</td>
<td>100</td>
</tr>
<tr>
<td>6</td>
<td>37</td>
<td>68</td>
<td>73</td>
<td>32</td>
<td>57</td>
<td>56</td>
<td>50</td>
<td>43</td>
</tr>
<tr>
<td>7</td>
<td>42</td>
<td>19</td>
<td>51</td>
<td>34</td>
<td>56</td>
<td>25</td>
<td>67</td>
<td>36</td>
</tr>
<tr>
<td>8</td>
<td>59</td>
<td>42</td>
<td>37</td>
<td>50</td>
<td>70</td>
<td>78</td>
<td>43</td>
<td>39</td>
</tr>
<tr>
<td>9</td>
<td>43</td>
<td>36</td>
<td>42</td>
<td>32</td>
<td>24</td>
<td>58</td>
<td>29</td>
<td>34</td>
</tr>
<tr>
<td>10</td>
<td>47</td>
<td>57</td>
<td>23</td>
<td>45</td>
<td>59</td>
<td>47</td>
<td>70</td>
<td>46</td>
</tr>
<tr>
<td>11</td>
<td>37</td>
<td>51</td>
<td>81</td>
<td>24</td>
<td>47</td>
<td>62</td>
<td>55</td>
<td>41</td>
</tr>
<tr>
<td>12</td>
<td>46</td>
<td>79</td>
<td>58</td>
<td>59</td>
<td>39</td>
<td>16</td>
<td>58</td>
<td>47</td>
</tr>
<tr>
<td>13</td>
<td>64</td>
<td>52</td>
<td>51</td>
<td>46</td>
<td>29</td>
<td>73</td>
<td>74</td>
<td>47</td>
</tr>
<tr>
<td>14</td>
<td>44</td>
<td>39</td>
<td>48</td>
<td>40</td>
<td>20</td>
<td>72</td>
<td>37</td>
<td>48</td>
</tr>
<tr>
<td>15</td>
<td>80</td>
<td>55</td>
<td>35</td>
<td>27</td>
<td>53</td>
<td>70</td>
<td>51</td>
<td>45</td>
</tr>
<tr>
<td>16</td>
<td>38</td>
<td>68</td>
<td>60</td>
<td>64</td>
<td>53</td>
<td>55</td>
<td>63</td>
<td>41</td>
</tr>
<tr>
<td>17</td>
<td>39</td>
<td>35</td>
<td>66</td>
<td>76</td>
<td>73</td>
<td>63</td>
<td>36</td>
<td>34</td>
</tr>
<tr>
<td>18</td>
<td>52</td>
<td>36</td>
<td>63</td>
<td>53</td>
<td>71</td>
<td>47</td>
<td>63</td>
<td>29</td>
</tr>
<tr>
<td>19</td>
<td>39</td>
<td>52</td>
<td>63</td>
<td>12</td>
<td>26</td>
<td>48</td>
<td>52</td>
<td>42</td>
</tr>
<tr>
<td>20</td>
<td>24</td>
<td>43</td>
<td>38</td>
<td>51</td>
<td>68</td>
<td>55</td>
<td>30</td>
<td>49</td>
</tr>
<tr>
<td>21</td>
<td>70</td>
<td>25</td>
<td>43</td>
<td>35</td>
<td>30</td>
<td>72</td>
<td>52</td>
<td>41</td>
</tr>
<tr>
<td>22</td>
<td>58</td>
<td>48</td>
<td>79</td>
<td>40</td>
<td>55</td>
<td>74</td>
<td>49</td>
<td>24</td>
</tr>
<tr>
<td>23</td>
<td>51</td>
<td>54</td>
<td>72</td>
<td>45</td>
<td>46</td>
<td>78</td>
<td>53</td>
<td>17</td>
</tr>
<tr>
<td>24</td>
<td>32</td>
<td>74</td>
<td>51</td>
<td>54</td>
<td>75</td>
<td>67</td>
<td>47</td>
<td>46</td>
</tr>
</tbody>
</table>

The b g $R$ or $R^2$ values for the data were quite close to zero indicating no linear relationship. Also, various functions do not appear to provide a very close fit for the data. Therefore, the manager tries the CTSP statistical test to see
if the product prices influence the sales of the their product and the substitute products also along with the weekly profit. So the MSMCO TSP shortest route algorithm is adjusted for nine dimensions (using the generalization of the Pythagorean theorem again to measure distances through nine dimensional space).

The algorithm found a shortest route of total distance 2999.856 on a complete tour connecting all 79 points in nine dimensional space. Then four random sets of \( n=79 \) \( X_1, X_2, X_3, X_4, X_5, X_6, X_7, X_8, \) and \( X_9 \) values (in the same range as the actual data) were inserted into the MSMCO TSP nine dimensional routing program. The four shortest routes produced from the random data were 5417.650, 5306.102, 5329.321 and 5263.210. Each of these four routes plus the route for the actual data used about two minutes of computer time on a 1999 desktop PC.

Table 4: The 2999.856 Route Point to Point

<table>
<thead>
<tr>
<th></th>
<th>49</th>
<th>61</th>
<th>57</th>
<th>79</th>
<th>55</th>
<th>4</th>
<th>22</th>
<th>23</th>
<th>35</th>
<th>13</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>28</td>
<td>10</td>
<td>36</td>
<td>7</td>
<td>18</td>
<td>26</td>
<td>20</td>
<td>2</td>
<td>27</td>
<td></td>
</tr>
<tr>
<td>33</td>
<td>15</td>
<td>8</td>
<td>3</td>
<td>70</td>
<td>50</td>
<td>77</td>
<td>58</td>
<td>69</td>
<td>30</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>32</td>
<td>6</td>
<td>11</td>
<td>31</td>
<td>9</td>
<td>29</td>
<td>14</td>
<td>21</td>
<td></td>
<td></td>
</tr>
<tr>
<td>37</td>
<td>60</td>
<td>46</td>
<td>75</td>
<td>53</td>
<td>73</td>
<td>48</td>
<td>78</td>
<td>45</td>
<td>63</td>
<td></td>
</tr>
<tr>
<td>51</td>
<td>43</td>
<td>67</td>
<td>42</td>
<td>52</td>
<td>54</td>
<td>47</td>
<td>44</td>
<td>17</td>
<td>38</td>
<td></td>
</tr>
<tr>
<td>24</td>
<td>16</td>
<td>25</td>
<td>39</td>
<td>25</td>
<td>34</td>
<td>40</td>
<td>72</td>
<td>64</td>
<td>59</td>
<td></td>
</tr>
<tr>
<td>71</td>
<td>66</td>
<td>41</td>
<td>68</td>
<td>76</td>
<td>62</td>
<td>65</td>
<td>56</td>
<td>74</td>
<td>49</td>
<td></td>
</tr>
</tbody>
</table>

Therefore,\n\[ CTSP = A/B = 2999.856/((5306.102+5329.321)/2) = .5641 \]
which is considerably less than 5263.210/5417.650 = .9715 which is the smallest of the 4 x 3 = 12 CTSP quotients (using the four random ones for A and B in CTSP under the hypothesis of no correlation between the nine variables). Therefore, the hypothesis of no correlation can be confidently rejected.

Although single valued functions don’t fit the data well, the relation equation (2)
\[
(X_1-50)^2 + (X_2-50)^2 + (X_3-50)^2 + (X_4-50)^2 +
(X_5-50)^2 + (X_6-50)^2 + (X_7-50)^2 + (X_8-50)^2 +
(X_9-50)^2 = 2500
\]
seems to be a pretty good fit for the data. The only problem with this “sphere” in nine dimensions is that setting the prices at \( X_1=X_2=X_3=X_4=X_5=X_6=X_7=X_8=50 \) produces the two extremes of profit \( X_p=100 \) (thousands of dollars a week) and \( X_p=0 \) dollars a week. Is this the optimal solution? The clear answer is yes some of the time and no the rest of the time!

The difficulty is (and this can happen in business and science) that there appears to be an extra, call it tenth variable if you will, that was left out of the model and must be accounted for by the manager. A look at the column nine profit data notes a dramatic decline in profit in the last 39 weeks the data was collected. This was when the chief competitors started an intensive price cutting and advertising campaign to capture more market share.

Thereore, the fitted relation could be viewed as two equations. The top half of the nine dimensional “sphere” for the first forty weeks, where the optimal solution is \( X_1=X_2=X_3=X_4=X_5=X_6=X_7=X_8=50 \) for the 50% mark up for prices with \( X_p=100 \) (thousands of dollars) in profit.

However, once the intensive competition started then the lower half of the eight dimensional “sphere” governs the profit and in an attempt to still maximize profit (at a somewhat lower figure) and keep the marks ups reasonably competitive, profit values of \( X_p=50 \) (thousands of dollars) on the “rim” of the lower half of the “sphere” should be the goal. One such solution for the manager (there are multiple ones) is \( X_1=X_2=X_3=X_4=X_5=X_6=X_7=X_8=32 \). Therefore, an average markup of 32% for the products should be used when the fierce competition happens. This will still keep weekly profit in the 50,000 dollar range.

Thereore, the competition will hurt the company’s profit, but the data (and subsequent fitted relation) say it will be less of a problem if the 50% mark ups are reduced to about 32% during tough competition weeks.

A key factor in any multivariate statistical test for correlation or “cause and effect” is to consider the possibility of a hidden relation (when the functional approach fails). This can happen frequently if a key variable has been left out of the initial data collection and equation model development. The CTSP statistic can perhaps help to sort through these difficulties.

A SECOND TEST OF HYPOTHESES

The previously presented statistical test, in the nine variable problem, used a range of 0 to 100 for each of the nine variables. This was done because the 79 x 9 = 711 coordinate points (taken as a group) are all numbers from 0 to 100 inclusive. The random shortest routes were clearly much larger than the 2999.856 value for the data. This led to an easy rejection of the null hypothesis of no correlation between the variables.

However, in practice, one might wish to use the nine (possibly different and smaller) ranges from each variable’s data points, to random sample for the shortest routes (in an effort to estimate the sampling distribution under the null hypothesis of no correlation between the nine variables).

Another look at our data reveals the following ranges for \( X_1, X_2, X_3, X_4, X_5, X_6, X_7, X_8, \) and \( X_9 \) of 85, 58, 72, 72, 69, 62, 72, 38 and 89 respectively. As an example, all of the \( X_i \) values are from 12 to 84. Therefore, we use a narrower range of 84 – 12 = 72 for the \( X_i \) random values (rather than 100, which was previously used).

Four random sets of \( n = 79 \) nine dimensional points (in these reduced ranges) were selected. Their shortest routes
(again using multi stage Monte Carlo optimization) were calculated to be 3689.266, 3701.118, 3643.876 and 3509.101. Therefore, CTSP = A/B = 2999.856/(3643.876 + 3689.266)/2 = 8.182 is still statistically significantly smaller than 3509.101/3702.118 = 9.479 (the smallest of the 3 x 4 = 12 CTSP quotients using our new random data in the reduced ranges). Therefore, the null hypothesis of no correlation can be confidently rejected even with this more statistically conservative approach to the data ranges.

GENERAL MSMCO

Multi Stage Monte Carlo optimization (MSMCO) is a general purpose simulation based optimization technique for multivariate problems in the computer age. Briefly, the idea is to take a random sample of several thousand feasible solutions and store the best one. This is called stage one. Then centered about this “best answer so far” several thousand more feasible solutions are looked at in a slightly reduced region. The best answer from this is stored and a stage three looks at thousands more sample answers, always centered about “the best answer so far”. This process is repeated for as many stages as necessary until the multi stage process sort of “funnels” into the optimal solution (or a near approximation).

However, for large optimization problems (such as the three and nine dimensional TSP problems used here for the CTSP calculations), one multi stage Monte Carlo simulation may not produce a very good answer. In those cases, outer loops do 10, 20 or 30 of these simulations over and over again and keep track of which coordinate values (subroutines in this case) keep recurring. These are then pinned down as the outer loops continue to “average” the early MSMCO solution tries, until a good answer appears. Also, the efficiency of the TSP MSMCO algorithm that finds the shortest routes (distances) connecting data points (rather than cities or customer locations) can be increased by adding the following to the simulation. Calculate the distance from each data point to each other data point and store it in an array. Then sort these distances so that the closest points to each of the points appear at the top of this array. Then focus the MSMCO TSP simulation in that area because that is where the shortest routes clearly are.

Multi stage works in other fields too, such as manufacturing (Conley 2003) and finance (Conley 1995).

CONCLUSION

The CTSP correlation coefficient was featured here on a three variable data set of n=119 points. CTSP was used to discover and prove a strong relationship between the variables, even though it could not be modeled with a function. However, the data did fit a spherical model nicely. Then a nine variable n=79 data point set was similarly analyzed and CTSP proved again a strong relationship, between the variables even though it was a non-functional one because of a hidden variable left out of the business optimization problem that gave rise to the data.

The CTSP correlation coefficient appears to be able to find linear, nonlinear and non-functional relationships for large data sets in our computer age. The research into it continues and is helped by the great increase in speed and capacity of small desktop computers for simulation work. The CTSP approach is being developed to help in the data mining process of screening large amounts of multivariate data quickly to see if any possible relationships exist between the variables represented by the data. It seeks to complement the standard linear analysis that is currently available.

REFERENCES


BIOGRAPHY

William C. Conley was born in Lansing, Michigan, USA and went to Albion College where he studied mathematics and received a degree with honors in 1970. He then earned a masters degree in mathematics from Western Michigan University in 1971 and an M.Sc. and Phd in mathematics and computer statistics from the University of Windsor in 1973 and 1976. He joined the faculty of the University of Wisconsin at Green Bay in 1977 and is now professor of business administration and statistics. The developer of multi stage Monte Carlo optimization and the TSP class of statistics, he is the author of five books and more than 185 publications worldwide. His favorite research problems are the shortest route or traveling salesman problems (TSPs) especially the ones adapted to higher dimension (k=3, 4, 5, 6, 7,...) He received the Founders Award for Faculty Scholarship in 2001. He is a member of the American Chemical Society and a senior member of SCS since 1994. He is a fellow of the Institution of Electronic and Telecommunication Engineers of India, a member of Phi Beta Kappa and a Michigan Scholar in College teaching. He was named to Marquis Who’s Who in America 2005 and Who’s Who in the World in 2006.
MODELING COLLABORATIONS AS COMPLEX SYSTEMS

Günther Schuh, Alexander Sauer, Sebastian Döring
Laboratory for Machine Tools and Production Engineering (WZL)
RWTH Aachen University
D-52074, Aachen,
Germany
E-mail: G.Schuh@wzl.rwth-aachen.de

KEYWORDS
Complexity, Collaboration, Industrial Networks

ABSTRACT

Currently, in the branch of production industry, there is an increasing tendency to collaborate. Although promising a better survival in the globalising market structures, those collaborations oftentimes fail due to a lack of problem-oriented comprehension concerning the required systems setup. This results in a very high failure rate that is near 50%. Although various models account for an individual company as an entity, there is a lack of complementary approaches that address the characteristics of enterprise networks. Therefore, new approaches for modeling both the core complexity drivers in reality and the inherent complexity of collaborative production systems are required. This paper wants to draw attention to the application of principles for complex systems from natural sciences to collaborative enterprise networks as socio-technical systems; the position is taken that this might yield the missing complementary approaches. This paper aims to exemplify an outline for the development of a Generic Model of Complexity (GeMoC). The GeMoC shall provide a problem-to-system match for collaborations in production industry. In the following the underlying concept for the development of this model will be uncovered.

COLLABORATIONS IN PRODUCTION INDUSTRY

Paradigm Shift

Today, industrial companies are challenged by a highly dynamic environment. Globalising market structures demand the flexible development and manufacturing of cheap high-quality products. Companies are forced to specialise in order to minimise product complexity and reduce production costs. The more partners are working on a single task, the more the complexity for each partner decreases (Fig. 1). Collaborations and maintaining relationships within these have become a major concern for managers in industrial companies. Although collaborations are assisted by modern information technology and data-communication, both the complexity of coordination and, as a result, the resources needed for the control load of coordination, increase. It is proceeded on the assumption that there is an optimal point of lowest total complexity with a corresponding number of partners, that is depending on the type of problem to be solved collaboratively.

As studies already revealed, this point is difficult to find in practice: currently, the failure rate of collaborative projects in manufacturing industry is near 50% (Wegehaupt 2004, Sauer 2005). Multiple reasons can be given for these results. Nevertheless, the under-estimation of the system’s complexity is chiefly responsible for this. Regarding complexity, a matching of the type of the networks system setup with the type of task to be solved is often ignored by managers. In the case that they do take a matching into consideration, a lack of knowledge about network complexity leads to devastating results. Reliable models and instruments for managers to handle the internal complexity of networks and the complexity of the environment are not available, yet. Moreover, there exist multiple unforeseen, emergent network effects in elasticity, controllability as well as overall network and production system behaviour, which increase the system’s complexity.

Figure 1: Interrelations in Collaborative Complexity

Up to date, managers try to cope with problems emerging from collaborations by applying tools and methods of network sciences. However, after more than ten years of research in the field of enterprise-networks the problems of production industry could not be sufficiently solved. New fundamental insights and vital progress in science are therefore required. To yield more than an incremental improvement, a shift of the underlying paradigm is essential (Kuhn 2003). With regard to organisational and management sciences, the complexity paradigm constitutes such a new paradigm for understanding collaborations and promises potentials for major progress when dealing with socio-technical systems.

Research Challenges

The emerging challenge for future management science is the development of models and methodologies for holistic complexity management in newly emerging organisational
structures. Therefore it is key to gain an interdisciplinary understanding of complex systems. The following core questions emerging from the field of management science shall help to cope with the emerging complexity: How can the complexity of the environment be reduced? How can the complexity handling capacity of the system be increased? According to the constructivist school of Watzlawick (Watzlawick 2005) and Foerster (Foerster 2005), the human mind constructs a model of the environment to cope with the complexity of measurable parameters. Consequently, this reduction of the perceived complexity leads to an increasing management efficiency and is therefore similar to the Soft Systems Methodology (Checkland 1981) and the Viable Systems Model (Beer 1959). According to Riedl (Riedl 2000), the human senses and mind are archaic instruments for coping with reality from less complicated and complex times. He postulates that now, with an increasingly complex environment, the capacity of the apparatus for stimulus processing is no longer sufficient to fully resolve the complexity of a situation. Consequently, any reduction and simplification of complexity aims at creating a manageable level for human beings in charge of controlling complex socio-technical systems. The degree of simplification chiefly depends on the amount, the interdependence and the behaviour of the systems’ inherent core complexity drivers. They represent both the underlying structure, which is representing the static dimension of complexity, and the behaviour of the overall system representing the dynamic dimension of complexity.

Another way to cope with the new challenges emerging from the environment – which results in an increasing overall complexity – is the improvement of the complexity handling capacity of both the collaborative system and its individual organisations. Hence, the challenge for complexity research in enterprise networks is the identification of adequate methods for system representation, the analysis of interdependence between core elements, and the specification of complexity drivers, accounting for the complexity emerging from the environment.

**MODELING APPROACHES**

**Modeling Collaborations**

The Construction of models is a key aspect for coping with our reality. In a way, human beings always act by using models that help to reduce complexity and to understand the incomprehensible, e.g. in physics or religion (Ludwig 2001). There are two basic groups of models: models for the attainment of scientific knowledge and models for applied aspects. The former models are more theoretically oriented, such as descriptive and explanatory models. The latter are more orientated towards pragmatic aspects, such as forecast or decision models (Wolf 1979, Rieper 1992). Both groups of models have to be considered in combination: description models can be regarded as a pre-condition for the appliance of decision models. In the following, modelling approaches from different disciplines that potentially contribute to a complexity based model of collaborations will be outlined.

After more than one decade of research in the field of network sciences there is still a lack of adequate theories, consistent paradigms and formal methods for modeling and representation. According to Wegehaupt, a main reason for this is that there is no explicit concept of networks and collaborations (Wegehaupt 2004). A promising approach to model and represent networks and collaborations is the differentiation between the underlying control mechanisms. Latest empirical surveys on successful networks (e.g. Bonabeau and Meyer 2001, Car et al. 2002, Dannenmaier et al. 2003) give strong hold to the hypothesis that only two different paradigms of control in networks exist (Figure 2): The first is the paradigm of guided networks which comprises features of hierarchical control in terms of first order cybernetics with the controller being a constituent element of the system. This type’s fundamental specifications are hierarchical networks and focal networks. Guidance is realised in guided network by explicit planning of interaction in advance to execution.

The second paradigm is the self-organised, organic network which is implicitly managed by Adam Smith’s invisible hand of an external context (Smith 1776), not being explicitly a control element of the system (second order cybernetics); one might differentiate between dyadic and triadic networks. Inherent to its character of local intrinsic-triggered interaction, selforganised networks can only be managed in an implicit way, which requires a non-deterministic coordination of activities and an active conditioning of the network’s context by establishing an effective rule setting channels network-activities towards a specific aim. Synergistic effects emerge from making the network’s entities acting congruently, holding up a necessary level of efficiency. Doing so, the network’s entities adapt their own complexity (i.e. activities, structure, behaviour) to the external requirements as parameters of their context. Due to this a global order emerges as a result of congruent local interactions (Stacey 1993). Depending on the type of problem one of these two paradigms is effective. It has been proven that all intermediary forms do either fail or evolve to either of the two forms over time. The rather constructivist approach of guided networks suits especially product development and production; the implicit management’s potentials are aligning ideation activities, procurement and service. For innovation processes creativity and effectiveness are more important than efficiency, which constitutes a paradigm for traditional technomorphous milestone-concepts. Hence, stability and instability issues of industrial networks might be driven by factors related to appropriate network control, although these driving factors have not been established, yet.

![Figure 2: Control Paradigms for Networks](image-url)
The guidance paradigm is well established in cybernetic approaches for system control, systems engineering or management cybernetics. Taking into consideration the different dimensions of complexity in socio-technical systems, the two network and control paradigms entail different types and patterns of complexity for tackling different types of collaborative problems. Ashby’s Law of Requisite Variety postulates that only complexity can absorb complexity (Ashby 1956). Abiding this requires a matching of the collaborative system’s variety (behaviour) with the complexity of the problem to be solved, and continuous efforts to increase the Complexity Handling Capability.

Modeling Complexity

With the activities of the Santa Fe Institute in the early 1980’s, the paradigm of self-organisation emerged and opened a new branch for the description and control of complexity (Jost 2004). With the increasing number of elements in artificial systems that are turned into net-like entities – their control became increasingly complex (Bar-Yam et al. 2003). As a result, the deterministic top down approach to systems control became inefficient, or even impossible, in particular for highly dynamic environments. It is assumed that in the field of complexity simple and comprehensible laws exist. The field of study for complex systems holds the assumption that the dynamics of complex systems are founded on universal principles that may be used to describe disparate problems ranging from particle physics to the economics of societies (Kauffman 1993). The development of complexity science means a shift in scientific approach having the potential to profoundly affect business, organisations and government. Complexity science strives to uncover the underlying principles and emergent behaviour of complex systems. Complex systems are composed of numerous, varied, simultaneously interacting agents. The objective of complexity science is to comprehend these complex systems whilst considering the questions: which principles govern their behaviour? How do they adapt to change? How can they learn efficiently? How can they optimise their own behaviour?

The term complexity can represent two meanings that are relevant to this research:

- As an expression of structure, mostly internally oriented; either being part of networks or of an individual system;
- As an expression of emergence, associated with new behaviour and complexity emerging from environment.

Internal complexity can be regarded as a set of design parameters, even though this has not been sufficiently defined in cybernetic approaches. To cope with emergence, different entities might develop different types of Complexity Handling Capability; under these conditions, balance will hardly be achieved. Only paradigms that address the dynamics of industrial networks and the environment will be chosen for an elaboration within the scope of this research. In an organisational context, complexity provides an explanatory framework of how organisations act, how individuals and organisations interact, relate and evolve within a larger social ecosystem. Complexity also explains the unanticipated consequences of interventions (Buchanan 2004). The intricate interrelationships between elements within a complex system give rise to multiple chains of dependencies.

The theory of complex adaptive systems being state-of-the-art in the field of self-organisation cannot be assigned to one particular field of science (Niewstad 1997). However, it has found its way into many adjacent disciplines, e.g. evolutionary computation, evolutionary biology and technology management. For science - due to its traditional, linear and deterministic programme that supports ideas of certainty and randomness - self-organisation, which is a general theory for complex systems, is considered the new paradigm and fundamental challenge. In adaptive systems that involve large numbers of entities, emergent, global behaviour that emanates from localised interactions is a critical concept. Understanding and shaping emergence may be essential for the success of such systems; from this perspective, explanations have been found that yielded more appropriate insights for phenomena that are difficult to comprehend.

Agent-Based Modelling is a new and special branch of computer simulation that emerged as a methodology for studying complex systems (Buchanan 2004). Agent-Based Models consist of agents, which have states and behavioural rules, and an environment. In the environment, which is either spatial (e.g., a rectangular grid) or non-spatial (e.g., an abstract trading community), interactions among agents take place. The interactions can either be direct where the action immediately changes the state of the partner, or indirect when the action changes the environment which, in turn, causes the partner’s state to change. Similarly, theoretical evolutionary biology has recently used game theories to explain and describe phenomena related to speciation. Especially, Adaptive Dynamics considers new approaches for the description of stability in populations (Geritz et al. 1997, Meszéná et al. 2001), thereby relating the development of species to state spaces. Traditional social sciences, especially classical economics, have very strong assumptions concerning the rationality of agents. Most Agent-Based Modelling use bounded rational agents that have only local, limited information, and limited ability and time to process that information, comparable to the real-life situation in industrial networks.

Both complexity sciences and network sciences are two sides of the same coin for future research in different disciplines. Only if a profound and interdisciplinary understanding of complex adaptive systems is gained, quantum leap improvements in handling and purposefully using these systems will be attained. In a close interaction of both approaches the fields can mutually benefit from each other’s experience, knowledge and approved solutions. Thus, the potential progress in both disciplines, complexity science and network sciences, may not only be additive but multiplicative.

Evolutionary Approaches

The progress in the science of complexity has also affected models in evolutionary biology. Especially, the models of developmental pathways and co-evolution deserve closer attention with respect to industrial networks. The responsiveness dictates that companies and industrial
networks have to anticipate on changes happening in the market domain and the domain of technology. These changes closely relate to evolutionary biological models that describe and explain the evolution of species. Evolutionary biology draws a clear distinction between mutation and selection, the two factors that determine the evolution of organisms and species. The most accredited models that describe the interaction between organism and environment are:

- The NK-model based on fitness landscapes (Kaufmann 1993);
- The Evolutionary Stable Strategies, application of game theories to the domain of biology (Meszéna 2001).

A preliminary study of evolutionary mechanisms and their meaning for organisational development reveals the importance of the criteria of sustained fitness, optimisation and mutation in order to reach a local optimum and evolvability. This means the capability to penetrate the new product-market combinations and disperse in combination with bifurcation processes (Dekkers 2004).

Several approaches exist in literature to describe the evolution of cooperation and collaboration. Doz (Doz 1996), for example, stresses that evolution of cooperation might be constrained by conditions of the inception of alliance and influenced by the collaboration process that consequently takes place. Larsson et al. (Larsson et al. 1998), for instance, propose two types of inter-organisational learning dynamics by using game theories. Both types describe the dynamics of the transparency and receptivity as a result of conditions. The first type of inter-organisational learning dynamics deals with possible barriers whereas the second type concentrates on empowerment. Furthermore, Meeuksen (Meeusen 2002) has added a base for a more profound model by connecting the approach of Larsson et al. to Kaufman’s NKmodel for fitness landscapes (Kaufman 1993). The further development of these models might yield even more adequate insights into patterns of collaboration within industrial networks.

Research in this matter has to link to evolutionary biological models. During the past decade, advances have been made in game theories, the descriptions of co-evolution, altruism, etc. within the domain of evolutionary biology. These advances can be transferred to the domain of organisations and networks (Dekkers 2004), yielding more appropriate models to describe collaboration; in turn this might lead to a higher effectiveness of collaborations and a more purposeful development of cooperation. Additionally, a more effective collaboration will result in adaptations by agents in networks to the dynamic environment.

**Socio-Technical Approaches**

According to Ulrich, General Systems Theory deals with the representation of systems and characterises organisations as open, dynamic, purposeful and productive socio-technical systems (Ulrich 2001). Several approaches of General Systems Theory exist, such as Maturana & Varela (Maturana and Varela 1980) or Beer (Beer 1965) and others, aiming at a specification of generic organisational concepts; Ropohl (Ropohl 1999) for example focuses on the integration of social systems and technical systems within three dimensions.

Therefore, he distinguishes three subsystems: The action system, the execution system and the goal setting system as the dimensions of the inner structure for any socio-technical system. The methodology which is known as the Delft School Approach - practiced by the Section Strategy, Technology and Entrepreneurship (Delft University of Technology) - designs organisational structures for socio-technical systems. The approach limits itself to equifinality (Dekkers 2004) by means of the exploitation of the steady-state model, for the modelling of recurrent processes, the related organelle structure model and the breakthrough model for processes of change (Veld 1998). The most important aspects of this methodology are the design approach and the application of a systems theory that were already exemplified during the 1970’s. None of these systems theories have been adequately implemented in the domain of networks yet.

The mentioned methodologies apply systems theories in order to model organisations from a cybernetic point of view, which matches the third system level according to Boulding (Boulding 1956, Figure 3), and they combine these theories with a socio-technical approach for the design of new organisational structures. Since organisations represent the eighth system level according to Boulding, the systems theories might require some further elaboration by means of the adoption of theories for complex systems, networks and biological models. Concerning this level, the validity of the design approach should be scrutinised. The design approach has the characteristics of static, one-time interventions, which tool and die shops have to avoid due to their severe effects on organisations. The review of other theories, such as complex systems theories, networks theories and biological models, can facilitate the identification of the structures of tool and die shops and their arrangement in networks, which is required for adapting to environmental changes and continuous change.

![Figure 3: The nine levels according to Boulding (Boulding 1956)](image)

Humam-influenced complex networks have common properties, which are hardly in line with existing cybernetic approaches. As a consequence thereof the lack of network-orientation within such systems theories becomes obvious when one considers the fact that most companies nowadays act in such networks - Here one must draw the conclusion that existing approaches remain hypothetically and are not capable of representing the reality of networking companies. One of these properties, the so-called small-world property, which is the most common of the specific properties, states that the average path length in the network is relatively small compared to the system size (Milgram 1967). Another property of complex networks is clustering, i.e. the increased
probability that pairs of nodes with a common neighbour are also connected. Therefore, increased efforts were made to identify other measures of complex (enterprise) networks (Fricker 1996). Perhaps, the most important one is the distribution of degrees, i.e. the distribution of the number of links between the nodes. It has been pointed out that several real world networks have scale-free distributions, often in the form of a power law. In these networks, a huge number of nodes have only one or two neighbours, whereas a couple of them are multiple-connected. The three specific properties mentioned hardly appear in the original systems theories such as the Applied Systems Theory (Dekkers 2002). While a number of models have been proposed to generate networks with different combinations of the three properties, each of these models describes a process that ends up in a network having the desired properties. Less effort has been devoted to the design of a dynamic system that would not only generate but maintain such a network. While there exist only few model approaches (Friedli 2000, Schwangerger 2000), most of them are based on the assumption that the system size or the number of links increases. Therefore, advances in network theories should focus on the dynamics of socio-technical systems accounting for the typical properties of complex networks.

**Approach To GeMoC**

The new paradigm for industrial networks requires an intense collaboration between the mentioned domains of natural science and management science. This provides the basis for an interdisciplinary development of a Generic Model of Complexity (GeMoC) in order to create a problem-to-system match framework for collaborative systems in production industry. The core of is approach is the application of principles of complex systems theory from various sciences to collaborative enterprise networks as sociotechnical systems. These networks must be understood as complex systems that can only assure their viability through adaptation in inter-organisational networks. In contrast to traditional organisational paradigms, the recent insights from complexity science into the field of collaboration in production industry will allow the full exploitation of the benefits and may be successful. Despite the great complexity and variety of systems, universal laws and phenomena are essential to their inquiry and understanding. Scientific endeavour is based on the existence of universality to a greater or lesser degree, which manifests itself in several ways. In this context, the study of complex systems as a new endeavour strives to increase the ability to understand the universality that arises when systems are highly complex. A study of universal principles does not replace detailed description of particular complex systems. However, universal principles and tools guide and simplify inquiries for the study of specifics. For the study of complex systems, universal simplifications are particularly important. Sometimes universal principles are intuitively appreciated without being explicitly stated. A careful articulation of such principles enables to approach particular systems with a systematic guidance that the studies of complex systems oftentimes lack. Research in this area of complex systems will contribute to the issues of interdisciplinary complexity definition (in order to bring different mental frameworks closer together), translate this definition into a model of problem complexity and imposed complexity (the topic of emergence), and a model of production system complexity (internal complexity). The paramount scientific objective aims at merging the two complexity perspectives of complex problems and complex socio-technical systems into one match-making Generic Model of Complexity (Figure 4). This will help to determine an accurate network paradigm for different types of collaborative problems.

![Figure 4: Methodology leading to the GeMoC](image)

The research questions to a high degree are at risk of being unanswerable in the mentioned context. The specification of quantitative complexity measures as a constituent controlling element to a holistic complexity management can therefore only be subject to subsequent research endeavours that rely on a strong basis of knowledge and understanding of complexity issues in collaborative systems.

**CONCLUSION**

What kind of correlation exists between type and complexity of the collaborative problem and the most suitable underlying network structure for solving it?

**Implication For Research**

With the field of complexity research still being a patchwork of scattered insights, a pragmatic and interdisciplinary approach holds the potential of yielding valuable insights into complexity modeling in today’s networked production industry. The most common approaches focus on the complexity of structures (mostly internal complexity) having a static character; this links to the most common system theories. In our view, the dynamic dimension of complexity, found in recent progress in various sciences, will fit the characteristics of industrial networks. The scientific objective is to provide a properly designed framework-of-thought for the (technological) implementation of complexity management infrastructures, which rely either on state of the art information technologies or on new insights about the architecture and characteristics of complex systems. Consequently, the objectives are to create a framework for complexity controlling systems for future network management tasks.
Industrial Implications

The implementation of this framework enables companies to react more flexible when market opportunities arise, thus increasing their competitive position, and to manage the networks they participate in more adequately. The adaptation to changing environmental conditions and the drive for innovation and fast product development will benefit from the research results. New paradigms for industrial networks will stretch beyond the traditional issues of trust, power, and supply chain management. Although the development of tools will be the next but one step, the results should guide companies’ management of network dynamics, their higher degree of specialisation, the development and implementation of technologies, and their development of appropriate long or short term relationships. This will reflect on both the optimisation of the supply chains and the speed of innovation and product development.

ACKNOWLEDGEMENTS

The authors would like to thank Rob Dekkers for his contributions to this paper.

REFERENCES

BIOGRAPHY

GÜNTHER SCHUH was born 1958 in Cologne, Germany. From 1978 to 1985 he studied towards his master degree in both mechanical engineering and economics at RWTH Aachen University in parallel. Subsequently, he was awarded the Ph.D. degree in 1988 after working as a graduate research student for Professor Eversheim. Moreover, he was chief engineer at the Laboratory for Machine Tools and Production Engineering (WZL) until 1990. He was also designated senior lecturer for Operations Management and Industrial Theory at the University of St. Gall (HSG) in Switzerland. With completion of his postdoctoral lecture qualification (habilitation) in 1993 he was appointed professor for Operations Management and became member of the directory board at the Institute for Technology Management (ITEM). Professor Schuh is founder and president of the GPS Complexity Management Inc., Software and Consulting Company in Aachen (Germany), St. Gall (Switzerland) and Atlanta (USA).

In September 2002 Professor Schuh was appointed Professor for Production Engineering at RWTH and succeeded Professor Eversheim both as chair holder of the chair for Operations Management and as director of WZL and Fraunhofer-Institute for Production Technology (IPT) in Aachen. Apart from undergraduate and graduate student teaching, he lectured in several national and international postgraduate and executive master programs at HSG (e.g. EMBA, MBE, KMU) and at DUXX (MBL), Monterrey (Mexico) for several years.

ALEXANDER SAUER was born in Coimbra, Portugal. He studied mechanical engineering and economics and obtained his degrees in 2002 at RWTH Aachen University. From 2002 until 2005 he worked as research assistant at the Laboratory for Tool Machines and Production Engineering (WZL) under the supervision of Prof. Dr.-Ing. Dipl.-Wirt. Ing. Guenther Schuh. Since 2005 he is chief engineer of the business strategy department at this institute.

SEBASTIAN DÖRING was born in Aachen, Germany in 1979. He studied mechanical engineering and economics and completed his degrees in 2005 at RWTH Aachen University. Since 2005 he has been working as research assistant at the Laboratory for Machine Tools and Production Engineering (WZL) under the supervision of Prof. Dr.-Ing. Dipl.-Wirt. Ing. Guenther Schuh.
MODELING A COMPLEX REAL-TIME SYSTEM

Aari Happonen and Jari Porras
Lappeenranta University of Technology
Department of Information Technology
P.O.Box 20, FIN-53851 Lappeenranta, Finland
E-mail: {happonen, jari.porras}@lut.fi

KEYWORDS
Modeling, Real-Time System, Multiprocessor environment, Capacity management

ABSTRACT
The aim of this paper is to analyze and report our study concerning a complex real-time multiprocessor system. The problem was to develop a model for an execution environment with complex input patterns from multiple input sources. The idea was that this model could be used both for short term and for long-term capacity predictions based on simulated load levels of the execution environment using business forecasts of the future as a basis for model parameters. The model development itself is performed with several iterations and is based on system traces from the real environment. We report the problems we faced in different phases of this project and the state of the project today as well as our plans for future work.

INTRODUCTION
In this paper, a development process for a simulation model for complex real-time system is presented. The goal was to produce a simulation model capable of predicting the load level of the target system under different input parameter conditions. Main emphasis of this paper will be in the model development process and in problems faced as the project evolved. Parameters and inputs were studied carefully as successful usage of this model depends on good input modeling (Leenin 1999). Measured data from the real production system were used for verifying the model.

COMPLEX REAL-TIME SYSTEM UNDER STUDY
The problem area considered in this paper is in modeling of a complex real-time system composed of several highly variable load sources and one very powerful processing environment. The processing environment in this system is a multiprocessor system with large number of processing engines and complex internal operation. The main emphasis in this work is in the modeling of the processing environment for capacity management purposes. Figure 1 gives an idea of the system under study.

Several different types of load sources generate tasks into the processing environment. Most of the sources are used directly by the end users but some systems are more or less automatic, e.g. factory systems. Nevertheless, all sources generate tasks with varying sizes, frequencies and more or less varying computational hardness. This makes the modeling task challenging. As the model is built by analyzing the system behavior the parameters of the load sources need to be known.

![Processing Environment and its Inputs](image)

The processing environment has a single entry point, i.e. gateway, for tasks to enter into the processing environment. Most of the load sources submit their tasks into processing environment through the gateway but few load sources exists that utilize the processing environment directly. In general, the gateway makes it possible to divide the one big task flow to the system to many smaller ones and to identify the general parameters for those flows. In (Leenin 1999) data collecting is divided to two different cases depending on the control researcher has over the collection process. In this case, neither of those general cases describes the measurement process accurately. Data is collected automatically by the system, but it was possible to suggest changes into the collection process by the researcher. In this particular study, it was possible to have some control over the data collecting process, which isn’t typically the case (Barton et al. 2002).

Gateway records the arrival time of the task and the time when the processing environment has finished the processing and returns the results back to the original source. The operation of the gateway creates a possibility to measure the interarrival and computational times taken by different tasks in the system. Together with the source information, it is possible to analyze the input patterns of different load sources and thus to find out the parameters of each load source. The gateway operation was traced to see which of the load sources has the major affect to the system performance and at the same time which of them should be
focused at the modeling. Analysis for the tasks generated by different load sources is presented in Table 1.

<table>
<thead>
<tr>
<th>Processing time</th>
<th>Users</th>
<th>Factories</th>
<th>Local</th>
<th>3rd party</th>
<th>Measurement</th>
<th>Other</th>
</tr>
</thead>
<tbody>
<tr>
<td>43.9 %</td>
<td>23.2 %</td>
<td>21.0 %</td>
<td>8.5 %</td>
<td>3.9 %</td>
<td>1.4 %</td>
<td></td>
</tr>
<tr>
<td>Message count</td>
<td>46.1 %</td>
<td>26.8 %</td>
<td>12.4%</td>
<td>8.9 %</td>
<td>1.8%</td>
<td>4.0%</td>
</tr>
</tbody>
</table>

Users, factories and local operation take most of the processing time as well as generate most of the tasks. Therefore, it was evident to concentrate the efforts to those three major load sources.

MODELING COMPLEX REAL-TIME SYSTEM

This project’s goal was to generate as general model as possible. To have a total control over everything, it was decided to use Matlab as a tool to build the models. In addition to the Matlab, some basic Linux GNU tools and own C/C++ programs were used to process the data gathered from the system environment under study.

As the goal is to model the behavior of the processing environment, both the environment as well as the behavior of the load sources needs to be known. One of the basic assumptions in the beginning of the model building process was that the measured processing time would indicate the amount of time one processing element is completely reserved from the processing environment by the task in hand and there would be no possibility for concurrent task execution. The modeling was started by tracing the behavior of the system in the gateway. The behavior of the load sources were analyzed by calculating interarrival times of the tasks and the task sizes. The system tracing process produced also the real empirical data to be used as input for the model to verify how changes in modeling process will affect to the results and to verify the accuracy of the model (Law and Kelton 1982). True system load is measured directly from the execution environment by using tools that are native part of the environment.

Model 1: Basic Queuing Model

Modeling of the processing environment was started with a very simple queuing model and with assumption that the processing of a task consumes 100% of the resources of one processing engine. To keep the model as simple as possible (Sadowski and Grabau 1999; Persson 2002), the modeling was started by combining all the inputs into one big input queue of the processing environment. Processing environment was simplified by multiplying the speed of one engine with the number of engines available at the processing environment. The general idea of the first model is presented in Figure 2.

![Figure 2: The first simulation model](attachment:image.png)

Figure 3 presents the results for the first model. In Figure 3, green line presents the real measured load of the system and the blue line represents the simulated results. It is obvious from the results, the first model is too simple to be able to produce the behavior of the true system. It was found out that the simulated load level grows too fast related to the mass of tasks entering into the system.

![Figure 3: Results of the First Simulation Model](attachment:image.png)

Model 2: Multiprocessor Queue Model

The simple model was further enhanced by considering multiple load sources and multiple processing elements as separate elements of the model. Results were not too encouraging. Actually, situation was nearly the same than with the first model (Figure 3). This problem was approached by analyzing the system queue levels in the simulation and in the real system. It was found out that in real system tasks arriving to the input queue were taken to processing nearly directly but in simulated system task had to wait in input queue with high probability. Studies into the real system revealed that the real system had a queuing system build inside the processing environment. This queuing system implements a 3-level queue structure with different queues for tasks entering into the system, tasks waiting to be served again after disk I/O and for the tasks taking long time to be processed. Therefore, the basic assumption of continuous task execution was busted.

Step 3: Priority queue model

The scheduler inside the processing environment works in three different levels. Medium level is for the new tasks entering into the system. High priority queue is for those tasks that are interrupted for a short moment caused by disk I/O. Low priority queue is for those tasks that take longer time to be processed. General idea is that the lower priority queues are processed only if there is no tasks to be processed waiting in higher-level queues. Implementation of the queues was quite easy (except the disk I/O). Problem with disk I/O queue is that there is no way to measure the time tasks wait in disk I/O queue in the real system. Thus, this time had to be approximated based on other measurements from the real system (for example using the cache-hit rate reported by the system). Conceptual design for the third model is presented in Figure 4.
Figure 4: Third Version of the Model

After the changes, the input queue waiting probability was in the same range in both systems. In addition, processing loads in the simulation remained in comparable level with the real system loads. Comparison of the simulation results and real loads is presented in Figure 5.

Figure 5: Results for the Third Model

ANALYZING THE RESULTS

Analysis of the results of the third model reveals three main problem areas. First problem is in start of the simulation run, which is probably because of the warm-up time. However, similar behavior can be seen later with the curve representing the same time of a day. Second problem is the spikiness of the simulated load. It seems that this problem is caused by the high priority queue waiting time, which is now under investigation. Third problem area is in middle of the simulation run where true load is quite much more than the simulated one. Some daily maintenance tasks were found that stress the system around the gateway. We are investigating the possibility to run those tasks in different way so we could measure them too.

Absolute percent error between model and true results for the first model is approximately 22% and for the third model 9%. With out the warm-up period and the maintenance tasks the error for the third model is only 7%. Analysis shows that the error could be greatly reduced if the spikiness of the model results could be smoothed out.

One long run simulation was made to verify that there would be similar problems in longer runs than with the moderate length simulations. Figure 6 shows that earlier problems were still present. Green oval at the start of the simulation run is the warm-up time problem. Black rectangle is the same spikiness problem as in earlier results and the purple ovals are result of an attempt to find out better parameters for the I/O queue wait time. Therefore, what can be said about these results is that there is still lot to do to develop this model.

Figure 6: Results for the Long Run Simulation with Tweaked Parameters

Results that are more precise are needed before continuing to the next phase, i.e. what-if case analysis with the model using varying input parameters. Measurements are already been made concerning the input behavior of the task sources. For example, interarrival time and number of the tasks entering into the system is quite linearly related to the number of users accessing the system. In addition, the number of users can be predicted for different periods of the day if the daily maximum can be predicted.

CONCLUSIONS AND FUTURE PLANS

In this paper, we have presented a modeling process for a complex real-time multiprocessor execution environment. The goal for this study was to develop a simulation model that can be used for predicting system load behavior in different situations. For now, we are still trying to solve problems concerning the information measurement from the real system. Main concern is if it is even possible to find out a way to measure the disk I/O distribution information for this system because of the amount of the disk activity. The idea for our future work is that people working with daily business cases predict how the business will change in future and this information is converted to changes of the inputs so simulation cases matching business case based predictions can be run.

REFERENCES

GENERALIZED BACKPROPAGATION THROUGH TIME FOR CONTINUOUS-TIME NEURAL MODELS IDENTIFICATION

Krzysztof Fjurewicz
Silesian University of Technology
Akademicka 16, 44-101 Gliwice, Poland
e-mail: Krzysztof.Fjurewicz@polsl.pl

KEYWORDS
Neural network, Parameter identification, Sensitivity analysis, Hybrid simulation.

ABSTRACT
Dynamical neural networks are frequently used as models of complex systems. Backpropagation Through Time (BPTT) is a standard technique for dynamical (recurrent) neural networks learning. It depends on “unfolding” in time of the original recurrent neural network and application of the classic backpropagation algorithm. There are similar approaches known in the literature avoiding the unfolding in time which may be very complicated for some neural-based models. Recently we propose the Generalized Backpropagation Through Time (GBPTT) algorithm for continuous-time neural networks identification based on discrete-time measurements. Here we present the proof of the correctness of the method.

PROBLEM FORMULATION
The classical scheme of the identification experiment is presented in Figure 1. The real complex system $S$ is modeled by the neural-based model $M$. Both, the real system and the model, are assumed to be continuous-time. Here we do not specify the topology of the model which may be chosen by the modeler. The system and the model are stimulated by the common and known continuous-time input signal $u_m(t)$. The initial and the final times of the identification experiment are $t_0$ and $t_f$ respectively. $p$ is a vector of weights of the neural model to be identified.

The output of the system $y_s(t)$ is measured at $N$ time moments: $t_n^N, t_n \in [t_0, t_f]$ giving discrete-time measurements: $y_s(t_n), n = 1, 2, ..., N$.

The identification task involves minimizing the following performance index

$$J = \sum_{n=1}^{N} h(e(t_n), t_n) \quad (1)$$

Where

$$e(t_n) = y_m(t_n) - y_s(t_n) \quad (2)$$

is the identification error at $t_n$ time, and $h(e(t_n), t_n)$ is a given nonlinear differentiable function. In most cases $h(\cdot)$ is a quadratic function.

Identification Task 1.

Find $\hat{p}$ such that the performance index (1) is minimized.

When a gradient-based optimization procedure is used to carry out the task above, then the following problem has to be solved:

Identification Task 2.

Find the gradient of the performance index (1) w.r.t. the vector of the weights $p$

$$\nabla_p J \quad (3)$$

Figure 2 presents the extension of the scheme from Fig. 1. There are additional elements and signals, $p_d(t_n)$ and $y_d(t_n)$.

Fig. 1. The block diagram of the identification experiment
One may write

\[
\begin{align*}
\dot{p}_d(t_0) &= p \\
\dot{y}_d(t_N) &= J
\end{align*}
\]  
(4)

Hence the gradient (3) reduces to the Jacobi matrix

\[
\nabla_p J = \frac{\partial y_d(t_N)}{\partial p_d(t_0)}.
\]

(5)

The Identification Task 2 now reduces to the problem of finding the sensitivity of the signal \( y_d \) at time \( t_N \) w.r.t. the signal \( p_d \) at time \( t_0 \).

The sensitivity function (5) may be obtained by using the sensitivity model presented in Fig. 3. The model is valid for variations of all signals (dashed symbols). \( \bar{M} \) is the sensitivity model of the model \( M \) and \( \dot{h}_e(t_n) \) is the derivative of the function \( h_e \) with respect to \( e \) calculated for different discrete-time moments \( t_n \).

Fig. 3. The sensitivity model of the system from Figure 2.

Finally, the sensitivity model may be presented as a serial connection of continuous-time (C-T) part and discrete time (D-T) part with the sampler and the pulser, where the input and output signals are discrete-time. See Fig. 4. In this scheme we use the symbol \( \bar{y}_c(t) \) instead of \( \bar{y}_m(t_n) = \bar{e}(t_n) \) to maintain homogeneity of notation.

\[ \bar{p}_d(t_n) \rightarrow \bar{y}_c(t_0) \rightarrow \bar{h}_c(t_n) \rightarrow \bar{y}_d(t_n) \]

Fig. 4. General scheme of the sensitivity model – the serial connection of continuous- and discrete-time parts.

The pulser produces one pulse only, at time \( t_0 \). The sampler samples the continuous-time signal \( \bar{y}_c(t) \) at times \( t_1, t_2, \ldots, t_N \) and produces the discrete-time signal \( \bar{u}_d(t_n) \).

The sensitivity model is always linear and may be non-stationary. In our case the sensitivity model has zero initial conditions. Hence the C-T part of the sensitivity model may be described by means of the integral operator

\[
\bar{y}_c(t) = \int_{t_0}^{t} \bar{K}_c(t, \tau) \bar{u}_c(\tau) d\tau
\]

(6)

where \( \bar{K}_c(t, \tau) \) is the kernel of the operator. Similarly, the D-T part may be described using the sum operator

\[
\bar{y}_d(t_n) = \sum_{m=1}^{n} \bar{K}_d(t_n, t_m) \bar{u}_d(t_m)
\]

(7)

with the kernel \( \bar{K}_d(t_n, t_m) \).

Now, let us find the \( i \)th element of the sensitivity function (5)

\[
\nabla_p J = \frac{\partial y_d(t_N)}{\partial p_d,i(t_0)}.
\]

(8)

We may use the sensitivity model for which (8) is reduced to the following

\[
\nabla_p J = \frac{\partial y_d(t_N)}{\partial p_d,i(t_0)} = \frac{\bar{y}_d(t_N)}{\bar{p}_d,i(t_0)}
\]

(9)

The easiest way to calculate (9) is to make the denominator \( \Delta \bar{p}_d,i(t_0) \) equal to one. Hence the system of Fig. 4. has to be stimulated by the signal

\[
\bar{P}_d(t_n) = I_i \cdot \delta K(t_n)
\]

(10)

where \( I_i \) is a column vector composed of zeros except for the \( i \)th element equal to 1. After such stimulation the output \( \bar{y}_d(t_n) \) at time \( t_N \) is equal to (8).

Let us calculate this value using formula (7)

\[
\bar{y}_d(t_N) = \sum_{m=1}^{N} \bar{K}_d(t_n, t_m) \bar{u}_d(t_m)
\]

(11)

The signal \( \bar{u}_d(t_m) \) appearing in (11) is the sampled output of the C-T part.
\[ \bar{u}_d(t_m) = \bar{y}_c(t_m) \]  
(12)

and can be expressed using (6)

\[ \bar{y}_c(t_m) = \int_{t_0}^{t_m} \bar{K}_c(t_m, \tau) \bar{u}_c(\tau) d\tau . \]  
(13)

Because

\[ \bar{u}_c(t) = I \cdot \delta_D(t) \]  
(14)

where \( \delta_D(t) \) is the Dirac pulse, so the equation (11) reduces to the following

\[ \bar{y}_c(t_m) = \bar{K}_c(t_m, t_0) \cdot I. \]  
(15)

Using (12) and (15) in (11) one can obtain

\[ \bar{y}_d(t_N) = \sum_{m=1}^{N} \left[ \bar{K}_d(t_N, t_m) \bar{K}_c(t_m, t_0) \right] \cdot I. \]  
(16)

In [Fujarewicz, Galusza, 2004] we proposed a method for construction of so called modified adjoint system if the original system is given in block-diagram form. The method consists in replacing all elements by their equivalents according to Table 1.

| Table 1 Rules for construction of the sensitivity model of the modified adjoint system |
|---------------------------------|---------------------------------|---------------------------------|
| Element of the original system  | Element of the sensitivity model | Element of the modified adjoint system |
| Linear c-t dynamical element    | \[ K(s) \]                       | \[ K(s) \]                       |
| Linear d-t dynamical element    | \[ K(z) \]                       | \[ K(z) \]                       |
| Linear static element           | \[ A \]                          | \[ A \]                          |
| Nonlinear static element        | \[ f(t) \]                       | \[ H(t) \]                       |
| Summing junction                | +                                | +                                |
| Branching node                  | +                                | +                                |
| Ideal pulser                    | \[ t_n \]                        | \[ t_n \]                        |
| Sampler                         | \[ t_n \]                        | \[ t_n \]                        |

Similar structural approaches to construction of adjoint systems have been previously proposed for discrete- or continuous-time systems in the literature. The original contribution of [Fujarewicz, Galusza, 2004] was that rules were proposed for any hybrid (continuous/discrete-time) system containing ideal pulsers and samplers in addition to typical continuous- and discrete-time elements. These rules have been formulated without proof.

Here we will prove that these rules are valid for the system of Fig. 4. Let us construct the system according to Table 1. The result is presented in Fig. 5.

![Fig. 5. The system created according to the rules of Table 1 based on the system of Fig. 4.](image)

The assignments of instantaneous values of input and output signals in Fig. 4 and Fig. 5 are as follows

\[ \tilde{\bar{y}}_d(t_n) \leftrightarrow \tilde{\hat{y}}_d(t_f - t_n) \]
\[ \bar{y}_d(t_n) \leftrightarrow \hat{y}_d(t_f - t_n) \]  
(17)

The modified adjoint system may be used in order to find the \( i \)th component of the gradient (9) sought, instead of the sensitivity model, thanks to following theorem.

**Theorem**

If the instantaneous values of all input signals in systems presented in Fig. 4 and Fig. 5 are equal to 0 except for

\[ \hat{u}_d(t_f - t_N) = 1 \quad \text{and} \quad \bar{\bar{y}}_d(t_0) = 1 \]  
(18)

then

\[ \tilde{\hat{p}}_d(t_f - t_0) = \bar{y}_d(t_N) \]  
(19)

**Proof**

The right-hand side of (19) is provided by expression (16). Now, let us calculate the left-hand side of (19).

As before, the modified adjoint systems are linear, time-variab systems with zero initial conditions, and they can be described by means of integral and sum operators respectively

\[ \hat{y}_c(t) = \int_{t_0}^{t} \bar{K}_c(t, \tau) \hat{u}_c(\tau) d\tau \]  
(20)

\[ \bar{y}_d(t_f - t_n) = \sum_{m=n}^{N} \bar{K}_d(t_f - t_n, t_f - t_m) \hat{u}_d(t_f - t_m) \]  
(21)

The left-hand side of (19) equals
\[ \hat{y}_{d}(t_f - t_0) = \]
\[ = \hat{y}_{c,d}(t_f - t_0) = I_T \cdot \int_0^{t_f - t_0} \hat{K}_c(t_f - t_0, \tau) \hat{u}_c(\tau) d\tau \]
\[ (22) \]

Since \( \hat{u}_c(\tau) \) is a series of Dirac pulses, we obtain
\[ \hat{p}_{d,j}(t_f - t_0) = \hat{y}_{c,d}(t_f - t_0) = \]
\[ = I_T \cdot \sum_{m=1}^{N} \hat{K}_c(t_f - t_0, t_f - t_m) \hat{y}_d(t_f - t_m) \]
\[ (23) \]

From (17) and (21) we may calculate \( \hat{y}_d(t_f - t_m) \) as follows
\[ \hat{y}_d(t_f - t_m) = \hat{K}_d(t_f - t_m, t_f - t_N) \]
\[ (24) \]

Substituting (24) into (23) we obtain
\[ \hat{p}_{d,j}(t_f - t_0) = \hat{y}_{c,d}(t_f - t_0) = \]
\[ = I_T \cdot \sum_{m=1}^{N} \hat{K}_c(t_f - t_0, t_f - t_m) \hat{K}_d(t_f - t_m, t_f - t_N) \]
\[ (25) \]

We will use the properties of kernels of (4) and (20), as explained in reference [Kailath, 1980]
\[ \hat{K}_c(t_b, t_a) = \hat{K}_c^T(t_f - t_a, t_f - t_b) \]
\[ (26) \]

and analogous properties of kernels of (7) and (21)
\[ \hat{K}_d(t_b, t_a) = \hat{K}_d^T(t_f - t_a, t_f - t_b) \]
\[ (27) \]

The right side of (16) may be rewritten as follows
\[ \sum_{m=1}^{N} \left[ \hat{K}_d(t_f - t_m, t_f - t_0) \hat{K}_c(t_f - t_m, t_f - t_0) \right] \cdot I_i = \]
\[ = \sum_{m=1}^{N} \left[ \hat{K}_d^T(t_f - t_m, t_f - t_0) \hat{K}_c^T(t_f - t_m, t_f - t_0) \right] \cdot I_i = \]
\[ = \sum_{m=1}^{N} \left[ \hat{K}_c(t_f - t_0, t_f - t_m) \hat{K}_d(t_f - t_m, t_f - t_N) \right]^T \cdot I_i = \]
\[ = I_T \cdot \sum_{m=1}^{N} \left[ \hat{K}_c(t_f - t_0, t_f - t_m) \hat{K}_d(t_f - t_m, t_f - t_N) \right] \]
\[ (28) \]

The result is equal to the right side of (25), which concludes the proof.
PHENOMENON COMPUTATIONAL PATTERN: COUPLING RELATIONSHIP BETWEEN PHENOMENA ON MULTI-PHYSICS SIMULATION

Felix C. G. Santos
José M. A. Barbosa
Eduardo R. de Brito Jr
Federal University of Pernambuco
Department of Mechanical Engineering
Rua Acadêmico Hélio Ramos, s/n
Recife - PE 50740-530 - Brazil
e-mail: fggs@demec.ufpe.br

KEYWORDS
finite element method, multi-physics simulation

ABSTRACT
Simulation of natural phenomena, such as the evaluation of temperature and air flow distribution inside a room, evaluation of material damage due to mechanical and chemical loads, is crucial in the daily life of engineers. This paper presents the Computational Phenomena Pattern, whose objective is to standardize - through a computational abstraction - the complex interaction of coupled natural phenomena in the context of the Finite Element Method. The pattern makes it intuitive and easier the representation of data sharing and dependence between different interacting phenomena when developing simulators based on the Finite Element Method. The pattern models natural phenomena data, the involved processes and their relationships, which are used in a typical simulation process.

INTRODUCTION
The success of Computational Mechanics is due to its effectiveness in solving problems that interest society and in providing deeper understanding of natural phenomena (facts, which occur in nature, like motion of material points and heat transfer in a continuum). Computational mechanics has a tremendous predictive power, making it possible the simulation of complex natural events and the further use of these simulations in the design of engineering systems [1]. However, in many aspects, it still applies software engineering development techniques related to the seventies. The consequence is that there are few reliable and effective tools for simulators development support and new achievements in numerical methods become difficult to introduce in the existing simulators. This work is part of an effort to overcome this difficulty in which concerns to the development of an environment - called PLEXUS - dedicated to the automatic development of multi-physics and multi-scale simulators based on the Finite Element Method [3]. We define multi-physics as a qualifier for a set of interacting phenomena, in space and time. These phenomena are usually of different natures (deformation of solids, heat transfer and electromagnetic fields) and may be defined by different scales of behavior (macro and micro mechanical behavior of materials). A multi-physics system is also called a coupled phenomena system.

THE FINITE ELEMENT METHOD
The Finite Element Method is a technique for the discretization of phenomena, whose behavior is determined by variables defined on a continuum (space and time). For instance, assume that a given phenomenon is defined by a function $u : \Omega \rightarrow \mathbb{R}$, $u \in \mathcal{H}$, where $\Omega \subset \mathbb{R}^n$, $n \geq 1$, is a bounded geometric domain and $\mathcal{H}$ is an appropriate space of functions. Suppose that $u$ satisfies a behavior law (a partial differential equation and boundary conditions, for example) $\mathcal{L}(u) = f$, where $\mathcal{L} : \mathcal{H} \rightarrow \mathcal{Y}$ is a differential operator and $f \in \mathcal{Y}$ is given. $\mathcal{Y}$ is a vector space and $\mathcal{Y}'$ its dual space. Based on that PDE, an equivalent integral formulation (weak form) is then obtained by a bivariate form $B_\Omega : \mathcal{H} \times \mathcal{Y} \rightarrow \mathbb{R}$ such that $B_\Omega(u, v) = L_\Omega(v)$, for all $v \in \mathcal{Y}$, where $L_\Omega : \mathcal{Y} \rightarrow \mathbb{R}$ is a linear functional. This is the initial setting for the finite element discretization techniques. Basically speaking, the major processes, which compose a simulation by the FEM are:

a) **Mesh generation process**: the building of an approximation, $\Omega_h$, to the exact geometric domain $\Omega$. $\Omega_h$ by the union of a set, $\tau_h$, of simple geometric entities. $\tau_h$ is the geometric mesh. Each simple geometric entity $e \in \tau_h$ is called a finite element.

b) **Discrete weak form process**: finite dimensional spaces $S_H$ and $S_Y$ are defined in such a way that an approximation to the original weak form can be defined as $B_\Omega(u_h, v_h) = L_\Omega(v_h)$, for all $v_h \in S_Y$, where $u_h \in S_H$ is the approximation to the exact
solution \( u; B_{Q\alpha}; S \times S_Y \) and \( L_{Q\alpha}; S_Y \) are approximations to \( B_Q \) and \( L_Q \), respectively. \( u_h \) is defined by a finite set of parameters \( U \), which are the basically unknowns of the problem. The discrete weak form represents a system (called global) of algebraic equations in \( U \). If the system is linear in \( u \), then the resulting global system will be a linear algebraic system, defined by a matrix \( K \) and a vector \( F \). In this case, it can be shown that \( K \) and \( F \) can be build using small matrices and vectors, which are computed based on restrictions of \( B_{Q\alpha} \) and \( L_{Q\alpha} \) to each finite element \( e \in \tau_h \).

c) **Solution algorithm process:** it means one among many ways of performing the desired simulation, that is, solving the global algebraic system of equations, obtaining the real vector \( U \).

d) **Visualization process:** this process comprises the computation of quantities to be visualized and the visualization procedures itself.

If the example described above is a part of a multiphysics system, it means that \( B_Q \) and \( L_Q \) may be dependent on vector fields from other phenomena. If that is the case, the computation of small matrices and vectors at the finite element level (item (b) above) depends on data from other phenomena - that is, it is coupled to other phenomena. Other type of data dependence is the case where two or more phenomena are defined on the same geometry component and share the geometric mesh.

The Phenomenon Pattern is concerned to:

i) The ways in which the discrete weak form components \( B_{Q\alpha} \) and \( L_{Q\alpha} \) restricted to a generic finite element can be computationally represented. This means that the Phenomenon Pattern provides abstractions such that: (a) all data pertinent to the computation of the small matrices and vectors at the finite element level can be represented; (b) all data dependence and sharing with respect to other phenomena can be taken into consideration in a standardized form.

ii) Those procedures contained in the Solution Algorithm process, mainly in what regards the way it demands a phenomenon the computation of small matrices and vectors and their assembling into given structures. Therefore, the Phenomenon Pattern presents an interface, which defines the ways it can be used by any solution algorithm during a simulation.

**THE CONTEXT**

We start this section with the description of the levels of procedures, which can be identified during the development of simulators using the FEM. They are:

- The finite element level:
  - Sub-level of the production of matrices and vectors
  - Sub-level of the error estimation
  - Sub-level of the post-processing

- The solution level, composed of:
  - Sub-level of the assembling of algebraic systems and its solution;
  - Sub-level of interactions, which articulate solutions of different algebraic systems;
  - Sub-level of loops and interactions involving progression in time and adaptation of models and discretizations.

The computation of small matrices and vectors (at each finite element) may be coupled with other phenomena, meaning that the computation of those quantities needs pieces of information from other phenomena. Those pieces of information are defined at the solution level. Therefore, changing solution algorithms may produce widely spread changes across both levels.

In order to provide a further insight, consider that a phenomenon \( P_i \) is able of computing a set of quantities \( \{Q_j(P_i)\}_{i=1}^{m} \). During the simulation each phenomenon has a fixed number of predefined states, which are represented by certain pieces of data computed during the many stages of a simulation. Assume that a given quantity \( Q_j(P_i) \), for some \( j \), is coupled to another phenomenon \( P_k \). The definition of a coupling means that a certain number of states of \( P_k \) are used in the computation of \( Q_j(P_i) \). Assume that \( P_k \) has the set \( \{St(P_k)\}_{i=1}^{m} \) as the set of all its states. \( P_i \) does not know a priori, which states of \( P_k \) are to be used in the computation of \( Q_j(P_i) \), until the solution algorithm determines it (at the moment it requires \( P_i \) to compute \( Q_j(P_i) \)). This is a choice, which may be changed whenever a different solution algorithm is used. Suppose that only one state is required for the computation of \( Q_j(P_i) \), that is, the state \( St(P_k) \), for some \( r \). This information is passed to \( P_i \) along with the demand to compute \( Q_j(P_i) \) and a data structure where it should assemble it. Firstly, \( P_i \) should retrieve the state \( St(P_k) \) and then it will able of computing \( Q_j(P_i) \) (on each finite element from the mesh) and assembling it into the given data structure. Secondly, the decision about the state to be used in the computation of a coupled quantity is made at a substantially higher level than the level of the finite element. Those two aspects produce the major difficulties, which are present in the development of simulators for coupled phenomena. However, there are still other requirements, which make the problem even more intractable: (i) couplings may occur only on a part of the geometric domain (for instance, on a part of the boundary) or it may occur.
dynamically, as in contact problems; (ii) two phenomena, which are coupled in a geometric component may not share the geometric mesh, meaning that the use of a coupled state by one phenomenon needs the transfer of that state from the mesh of the coupled phenomenon to the mesh of the former.

The described difficulties generate a problem that can be phrased as: What abstraction can adequately represent and encapsulate the information, relationships and processes pertaining to the finite element level and which are concerned to the numerical modeling of a natural phenomenon, in order to describe and implement phenomena abstractions in the context of coupled multiphysics systems? The next sections present a solution to this problem.

THE PHENOMENON PATTERN

For the purpose of simplicity of explanation, a phenomenon is considered as a machine for computing small matrices and vectors and assembling them into given large matrices and vectors. The computation of these small matrices and vectors is performed on each finite element of a mesh. The following requirements have to be satisfied:

i) The computation of small matrices and vectors may be related to any geometric component among the set of geometric components, which define the physical domain of a phenomenon.

ii) Each phenomenon should have a list of quantities, which it is able of computing.

iii) The definition of the coupled states needed for the computation of a certain quantity of a phenomenon should be defined only at the solution algorithm level. A phenomena should not be aware of the way it is used during a particular simulation.

iv) Two phenomena may share meshes on any common geometric part.

v) A phenomenon is coupled to vector fields not to phenomena. A specific phenomenon is later on given as the owner of the specific coupled vector field.

A solution, which could satisfy the above requirements, is the Phenomenon Pattern (called Phenomenon from now on). It can be viewed as a container with two acyclic graphs: the PhenGraph and the GeomGraph (see Figs. (1), (2), (1) and (1)). Both of them have exactly the same structure as graphs, but the pieces of data stored in each GraphNode are different from one graph to the other. We call by PhenNode and GeomNode the GraphNode’s from the PhenGraph and GeomGraph, respectively. In GeomGraph the geometry data is stored in a brep structure (boundary representation). That is, from the root of GeomGraph to its leaves one goes from the geometric entities of higher dimension (volumes, for instance) to the lower dimension ones (points). On the other hand, in PhenGraph one finds in each PhenNode a set of procedures, which are to be computed on the geometric entity of the respective GeomNode.

Figure 1: Geometry Graph

Figure 2: Phenomenon Graph

Phenomenon also contains an indexed set of quantities \( Q = \{ Q_i \}_{i=1}^n \) (see QTable in Fig. (1)), which represents the set of all quantities (small vectors and matrices) that it is able of computing and assembling into given structures (large matrices and vectors). Here \( n \) is the total number of quantities the current Phenomenon can compute and assemble. The procedures responsible for the actual computation of those quantities should be stored in the PhenNode’s. This further implies that each PhenNode, say the \( j^{th} \) one, should have an indexed set \( q_j = \{ Q_{jk} \}_{k=1}^{n_j} \subset Q \) (see qTable in Fig. (1)), which represents all quantities the PhenNode \( j \) can compute.

Figure 3: Phenomenon-Geometry relationship
Each PhenNode needs some pieces of data in order to be able of computing a certain quantity. Those pieces of data are the following:

a) General Data (it serves all quantities):

- GeomMesh: it is the mesh of a the geometric entity stored in a GeomNode. It is a data structure, which contains all geometric finite elements. It belongs to the respective GeomNode.

- PhenMesh: its is the mesh of the phenomenon. It is a data structure, which contains information about the approximation of the phenomenon’s vector field on each geometric finite element.

b) Specific Data for each quantity \( Q_{jk} \), \( k = 1, \ldots, n_j \):

- CoupledPhenNodes \( J_{jk} = \{C_{jk}^{[x]}\}_{x=1}^{n_{jk}} \): it is the set of GraphNodes pertaining to other phenomena. They are used in order to obtain data from other phenomena, which are needed in the computation of \( Q_{jk} \).

- CoupledStates \( S_{jk} = \{S_{jk}^{[x]}\}_{x=1}^{n_{jk}} \): it is the set of states, which should be retrieved from the respective CoupledGraphNode \( C_{jk}^{[x]} \).

Next we explain how objects Phenomenon are used. The simulator is here considered as a pattern [2], [4], [9], which is - simply speaking - a workflow in the form of a tree and divided into four layers (see levels of procedures defined in Section(i)):

- Kernel: set of procedures related to algorithmic structures for the control of loops and interactions involving progression in time and adaptation models and discretization;

- Block: set of procedures related to the articulation of solutions of different algebraic systems;

- Group: set of procedures related to the assembling and solution of algebraic systems, together with the execution of operations with matrices and vectors.

- Phenomenon: encapsulates the set of procedures related to the production of small matrices and vectors and to their assembling in given larger data structures. It also performs other computations related to post-processing and error estimation.

The simulation starts with the execution of the root of the Kernel, which uses services provided by a set of Blocks, which in turn uses services from a set of Groups. Each Group owns a set of Phenomenon objects, which are used to compute small matrices and vectors and to assemble them into given (by its Group) larger data structures. The structure of Group classes (see Fig. (i)) has the following characteristics:

- It has a set of indexed data structures (see DataStructures in Fig. (i)), which can be shared upon demand by its own Phenomenon objects.

- It has a set of tables, which can be dynamically programmed in order to convey information about the computation of each matrix or vector: (i) the Phenomenon objects, which contribute to it; (ii) for each Phenomenon, the quantities, which are to be computed; (iii) the coupled states (from the respective coupled Phenomenon) to be used in the computation of each quantity produced by a Phenomenon object.

Therefore the Groups level is the level where definitions about the coupled states take place and are conveyed to each Phenomenon object during a demand for the
computation of a certain quantity. The coding of the software components for a Group class needs previous knowledge about the set of indexed data structures from all other Group classes. A Group object (call it G) demands a Phenomenon (call it P) object to compute a quantity in the following way:

- G is asked (by its Block) to compute a certain quantity $Q_g$.

- G retrieves from one of its tables the references to all Phenomenon objects, which contribute to $Q_g$. P is among them.

- G retrieves from another table a set of data for each contributing Phenomenon object. This set of data for the $i^{th}$ Phenomenon, which contributes to $Q_g$; (II) for each $q_{i_j}$, a set $CS_{ij} = \{S_{ij_k}\}_{k=1}^{n_{ij}}$, containing the coupled states to be used in its computation - the order in which those states are given is important, since it may be from different phenomena: (III) a reference to a data structure (say $K_g$) where the quantity should be assembled.

- Suppose, now, that P is given a demand to compute a certain quantity $Q_r$, together with a set of indexes for the coupled states, say $CS_{pr} = \{S_{pr_k}\}_{k=1}^{n_{pr}}$ and a reference to $K_g$.

- Then, P transfer the demand to its PhenGraph, which in turn send it to its root (say R).

- R checks if it is able of computing the desired quantity. If so, it sends the demand to one of its objects (say W), which is responsible for the computation and assembling of $Q_r$. R sends, together with the data already defined, a set of references, say CG, to the respective GraphNodes from the PhenGraphs of the coupled Phenomenon objects.

- W retrieves the set of coupled PhenMeshes (say, CPM) and the respective set of discrete vector fields (say VF) from the respective set of coupled GraphNodes. This is done by sending a request for the states contained in $CS_{pr}$ to the respective GraphNodes contained in CG. Each coupled GraphNode from CG is able of asking its owner Group for the desired state and sending it back to W.

- W traverse the PhenMesh (given by R) together with the coupled ones (contained in CPM) and is able of computing, for each finite element, the quantity $Q_r$ - using the coupled discrete vector fields (contained in VF) - and of assembling them into $K_g$.

- If R is not able of computing the desired quantity, or if it has already computed it, it will transfer the demand recursively to its children GraphNodes together with the same data it received. The process goes on recursively, until there are no children GraphNodes to be reached.

CONCLUSIONS

We have presented a pattern called Phenomenon in order to cope with the difficulties found in the development of simulators for coupled multi-physics problems. It was identified that the solution algorithms employed by the simulators provide data at a higher level, which determines many procedures at a lower level. Thus, small changes in the solution algorithm could generate the need for widespread changes along the simulator software, implying in strong reprogramming. A separation of concerns was important to identify that the simulator could be divided into four well defined hierarchical levels. The part of a solution algorithm where pieces of data are generated to be used at a lower level was concentrated at a level called the Group level. The part at a lower level where those pieces of data are used was identified as the Phenomenon level. An abstraction for the computational representation of the Phenomenon level was described as a pattern. It was shown that his pattern (Phenomenon Pattern) represents adequately all data pertaining to the discrete behavior laws of a phenomenon (production of small matrices and vectors), together with the needed sharing and transfer of data between Phenomenon objects and between Phenomenon objects and the solution algorithms. An important consequence was that all information and procedures, which are very specific of a solution algorithm (less reusable) are encapsulated in the Group level, while all information and procedures, which are strongly are encapsulated in the Phenomenon level. Experiences with prototypes have shown a tremendous improvement in: (i) the time spent in the development of simulators; (ii) the reusability of software components, opening the way for the building of repositories; (iii) the correctness of software components. References of previous and related work can be found in [4]-[9].

REFERENCES


Pattern Languages of Programming, Pernambuco, Brazil, 2003.


MANUFACTURING PROJECT DATA SIMULATION
IMPROVING MULTIPLE PROJECT MANAGEMENT USING SIMULATION

Torsten Licht, Christopher Schlick, and Holger Luzak
Chair and Institute of Industrial Engineering and Ergonomics
RWTH Aachen University
D-52062, Aachen, Germany
{T.Licht; C.Schlick; H.Luzak}
@iaw.rwth-aachen.de

Lothar Dohmen
Bayer Business Services GmbH
D-51368 Leverkusen, Germany
lotharwerner.dohmen@bayerbbs.com

Ludger Schmidt
Research Institute for Communication, Information Processing and Ergonomics (FKIE)
D-53343 Wachtberg, Germany
L.Schmidt@fgan.de

KEYWORDS
Multiple Project Management, Research and Development, Decision Support Systems, Person-Centered Simulation, Petri Nets

ABSTRACT

When managing multiple projects with various task and resource interdependencies it is challenging to plan task assignments for personnel and to derive a realistic schedule. Existing models for simulating the organizational processes of product development projects have emerged, but are of no adequate help for project managers yet.

In this paper a person-centered, actor-driven simulation approach is presented that can cope with highly parallel tasks and projects. The task execution sequence is determined by means of a bounded rational choice model accounting for the urgency of tasks. The simulation model considers task durations as output variables consisting of efforts spent and idle periods. It therefore naturally accounts for resource-induced interdependencies between tasks. The model creates a very realistic project dynamic using only a small number of input parameters. The project manager is, for instance, not forced to specify fixed predecessor-successor-relations. The most sensitive input parameters are the efforts (man hours) to be spent and the deadlines of tasks.

A case study of a real product development project was conducted along with the model. Using Student’s t-tests the simulated project durations showed no significant derivation from the empirically observed duration.

INTRODUCTION

Product development projects nowadays are carried out according to the Concurrent Engineering paradigm. As a result, the number of persons involved in product development projects and the number of concurrently performed projects permanently increases. The process of managing such parallel projects is often referred to as multiple project management. When the underlying tasks use the same resources (personnel, tools) the management of such multiple projects is quite challenging.

Research on Project Management provided many methods, for example, the well-known Gantt-Charts, the Critical Path Method or Program Evaluation and Review Technique. These methods are nicely integrated into standard project planning software like Microsoft Project.

With respect to the challenges of Multiple Project Management though, these methods and tools provide effective support only in theory. Taking a closer look at them it turns out that their modeling and prediction capabilities for complex development projects are severely limited. For instance, it is not possible to model and simulate iteration loops and resource dependencies adequately.

Furthermore, probability distributions of task durations are considered to be independent, which is obviously untrue for closely interacting development tasks. Last but not least, dependencies of the time-on-task given resources - like computer-aided design tools - are not considered.

In conclusion, the classical methods of project planning have significant shortcomings when modeling and simulating dynamic and concurrent product development projects.

EXISTING SIMULATION APPROACHES

These shortcomings strongly stimulated research in simulation of complex product development projects since the 1990s. Simulation models of product development processes can be distinguished through two approaches: 1) actor-driven simulation models; here, the unfolding system’s behavior is generated by actors (persons or organizations) according to some tasks (work to be done), 2) activity-driven simulation models; here, the unfolding system’s behavior is generated by activities consuming time and using some resources (persons, tools). The model presented in this paper belongs to the actor-driven models.

Related Work

Levitt et al. (1999) developed an actor-driven simulation model, the so-called Virtual Design Team (VDT) that incorporates Micro Contingency Theory to the information processing behavior of simulated actors. The VDT also accounts for different goals of actors, dynamical assignment of resources to activities, exception generation and exception handling. However, the VDT described so far is only capable of modeling one project team and one project at a time. The model described in this paper is capable of simulating an arbitrary number of parallel projects. Regarding the choice between concurrent parallel tasks, an actor stochastically applies specific priority rules, such as FIFO or LIFO, or a person randomly selects a task. Deadlines or personal goals are not considered for decision making. The authors do not provide validation studies where simulation results were compared to data of real projects.
Independently from Levitt’s group, Steidel (1994) developed another actor-driven simulation model of product development projects using procedural queuing simulation software based on the C-language. Simulated persons play an active role in this approach by checking if a valid job is available and then processing this job in case of available resources. When more than one job is available persons decide which job to process first. However, priority rules for these decisions are based only on a simple extension of the FIFO concept. Persons in this approach do not have a specific goal system – only capabilities in the form of qualification describing parameters are represented. Once a person is performing a task he or she will complete it without interruption. A shift of attention between tasks to be performed concurrently is not possible.

An activity-driven, Petri Net based approach for modeling work processes in engineering design within the Concurrent Engineering concept was described by Kusiak and Yang (1993). This model accounts for resources as well as serial and concurrent task sequences – including iteration loops. The transitions, however, are neither timed nor is it possible to assign probabilities to transition firing. The model can be used to assess the order of tasks and the assignment of resources in a project. Since task durations cannot be modeled, though, no project duration can be obtained from a simulation.

Raupach developed another activity-driven approach for simulating product development processes (Raupach 1999). In this approach three partial models were distinguished: 1) the model of activities, 2) the model of resources and the organization as well as 3) the data model. The model is capable of simulating and evaluating the consistency between different design solutions including a variant management. However, a person in this model is considered as a resource of an organizational unit – and therefore has no active role. Parallel performance of activities is not possible.

Browning and Eppinger (2000) describe the Design Structure Matrix (DSM) that enables easy-to-use definitions of predecessor-successor relations of design tasks. Based on the DSM they developed an activity-driven simulation model that uses rework probabilities, impact of rework and learning effects. The simulation, however, assumed unlimited resources and was not capable of simulating more than one project in parallel. The input data were provided from a case study of the development of an uninhabited aerial vehicle (UAV).

Cho and Eppinger (2005) developed another activity-driven simulation model that uses the DSM but accounts for limited resources. In case of two activities competing for one resource the model applies heuristic rules. An activity, for instance, gets a higher priority, if the rework-adjusted duration of that activity is higher than that of the other activity. By applying this rule those tasks that are on the “critical path” are to be processed with higher priority. This refers to the goal of minimizing the duration of a single project. In multiple project practice, however, such goals will be relaxed towards achieving a reasonable performance of the entire project portfolio. Additionally, with the approach of Cho and Eppinger, activities cannot be split and must therefore be completed in one step. It is not possible that two or more tasks can be processed in parallel by one resource. That does not correspond to real product development processes. Furthermore, task durations are considered as independent input variables of the simulation model. With the approach of Cho and Eppinger, in theory, a simulation of parallel projects is possible, but the authors only provided a simulation example based on Browninger’s UAV case study. The assumptions of the simulation model, such as the propagation of rework through a product development project, seem to be reasonable and consistent but are not validated by empirical findings. Project managers using DSM-based simulation models face another significant problem: the estimation of rework probabilities and rework impacts is a very complex task. Depending on prior knowledge and experience these estimations can dramatically vary between different project managers. Since these probabilities are sensitive input variables, the validity of the simulation results shows a corresponding variance.

An event graph based, activity-driven simulation model of a design process of a semiconductor fab is described by Gil (2001). The model focuses on the impact of changed design criteria on overall project duration considering different postponement strategies (retarding the start of the succeeding activity after the predecessor has finished). The model assumes deterministic durations as input variables, independency of activities and unlimited resources.

Kao et al. (2002) developed a multiagent framework for supporting concurrent engineering projects. Several agents were modeled with IDEF0 and High Level Petri Nets such as the Coordination Agency, Scheduling Agency or an Agent for a team member’s behavior. One of the features presented in this work is a simulation model of the concurrent engineering project. The model described is a mixture of actor-driven and activity-driven elements. The team members of a product development team are modeled as High Level Petri Nets. They can receive messages, execute tasks, reply to requests, deny the assignment of a task, be in an idle state, etc. The project model (task net) is modeled as a Petri Net where tasks are sub modules consisting of two states (“waiting to be executed” and “being executed”). The input conditions (e.g. team member is “available”) as well as all predecessor-successor-relations are modeled by arcs and some additional places. The resulting task net is therefore quite confusing. Task durations are used as input variables and estimated overall project duration is an output variable of the simulation model. The authors do not clearly describe what parts of their framework have already been implemented in a software model. It is also not clear how mixing the actor driven approach (team member model) and the activity-driven approach (project model) results in a consistent model. It remains open if the actors drive the system’s behavior or if the activities do. Even though the model is very overarching and complex it is not capable of modeling concurrent, parallel tasks carried out by one person. In addition, multiple projects with common resources cannot be modeled and the work break down structure is restricted to clearly bounded design tasks without stochastic characteristics. The approach of Kao et al. has not been used for simulating a real product development project.

Yan et al. (2003) present an activity-driven simulation model of product development processes using “Extended Stochastic High-Level Evaluation Petri Nets” (“ESHLEP”). The model’s underlying assumptions about inter-
dependencies of tasks are based on the DSM and on the IDEF3 model. The model uses many rules for activity sequencing, resource assigning, state changing and simulation termination. In order to choose between concurrent, parallel activities the model assigns the highest priority to the task with the smallest slack. According to the other DSM-based simulation models, the activity durations are independent input variables. It is not clear if also parallel projects can be simulated using this approach.

Considering this variety of existing approaches one could ask: Why should we develop yet another simulation model for product development projects? What are the biggest shortcomings of the existing models from a practical, multiple project point of view?

**Parallelism And Overlapping**

Tasks in a real project are almost never processed in true sequence by experienced design engineers. Usually a „successor“ task will be started when the „predecessor“ is done by, let’s say, 20 percent. This semi-parallel or parallel task processing cannot or only insufficiently be modeled with existing simulation approaches. Some models allow for defining semi-parallel or parallel tasks (e.g. Cho and Eppinger 2005), but during a simulation run one task will be completed once it has been started. That is, the observed task processing finally is done sequentially.

Due to this lack in parallel task processing the project manager is forced to define the work brake down structure on a very fine-grained level. In real product development projects however, it is impossible to define such fine-grained predecessor-successor-relations of development tasks at the start of the project. Therefore, these models may hardly encounter acceptance by project managers.

The VDT (e.g. Levitt et al. 1999) provides an exception. Here, tasks are decomposed into subtasks, which then can be processed in parallel.

**Resource-Induced Task Dependencies**

One of the major shortcomings of existing simulation models is that they – similarly to the classical project management techniques – assume the task durations as independent of each other. Some approaches consider information-induced interdependencies resulting in iteration loops. In these models task durations will be increased when an iteration occurs. But, from the resource point of view the task durations are assumed to be independent of each other. For predicting real multiple projects however, it is essential to account for resource-induced task interdependencies. One may for instance consider a person in a real project processing a “task A” and in between he or she has to finish some other tasks (may be of other projects). Due to the emerging idle period the duration of „task A“ will increase respectively.

Furthermore, most of the existing models are not capable of simulating two or more projects with parallel access to common resources and are therefore not helpful for managers of multiple projects.

**The Input Redundancy Problem**

Some of the existing models consider the workload assigned with a task (or respectively the task duration) as input and as output variables and, therefore, may double count some input information. Project managers that try to plan their projects with these models have to initially estimate the expected workload (respectively: duration) for a particular task as input variable. This estimation contains all knowledge and experience of the manager, including his or her judgment about rework and information interdependencies with other tasks. The models now assume that the managers tend to underestimate information interdependencies between tasks. Therefore, the same project manager has to estimate these interdependencies, which in turn influence the simulated workload (respectively: duration) causing rework or iteration loops.

The catch with these models is that in providing the “first guess” about the workload (respectively: duration) the project manager would have to exclude his or her knowledge about the information interdependencies, which is virtually impossible. An excellent, experienced senior project manager may in his or her first guess intuitively consider all relevant information interdependencies making these existing models redundant.

One possible solution to the problem could be provided if we were able to model workload or task durations as “completely dependent output” variables – without specifying a first guess.

At this point another weakness of the existing approaches becomes obvious. Workload does not solely depend on information interchanges but also on the complexity of the product to be developed, the degree of innovation, the preliminary knowledge of methods that can be used for solving particular design problems, the degree of experience with similar tasks, etc. Due to this rich variety of influencing parameters it would be hard to parameterize such a model.

We therefore suggest to consider the expected workload (efforts to be spent) as given and to model the task duration as “completely dependent output” variable – consisting of the efforts spent and idle times.

**Benefit For Multiple Project Management Practice**

The aim of existing simulation models of product development projects can be distinguished through two categories: 1) models that try to provide a most efficient project schedule (e.g. minimize development time, maximize project performance) by varying task sequences, communication structures or probabilities of information changes, and 2) models that try to estimate the schedule risk or the quality of results given some parameter settings.

The former models are partially based on classical methods of project management, focusing on rational, rather hypothetical measures (Yan et al. 2003, Cho and Eppinger 2005). These models overestimate the possibilities of influencing parameters in real projects to some extent. At the same time, they tend to underestimate the bounded rationality of real projects with sociological and psychological effects on work performance. The identified “most efficient” project settings and the respective implications for project managers should always be
scrutinized carefully. For a real project manager it is, for instance, not obvious how to “increase the degree of overlapping” of two concurrent tasks by 20 percent while the “information uncertainty” shall be decreased by 10 percent.

The model presented in this paper belongs to the latter ones. That is, the aim is to support project managers in assessing the estimated course of a product development project. The existing models in this category, however, also tend to underestimate the bounded rationality of real projects. Resource allocation rules or scheduling rules, for example, do not take into account that product developers act with bounded rational behavior (Kahneman 2003). Therefore, the simulation results from these models can only be poor representations of real product development projects.

Model Validation

Finally, existing simulation models are often poorly validated through qualitative or theoretical considerations. A comparative consideration of simulation results and empirically observed project data can rarely be found in literature. Some authors have modeled and simulated real projects, but since the task durations were used as input parameters it is not surprising that the simulated output (project duration) can be “predicted” satisfactorily.

NOVEL APPROACH

These reasons were sufficient to develop a new and improved simulation model for the management of complex development processes. The simulation model consists of the five partial models depicted in the Figures 1 to 5: 1) the partial model of the product to be developed, 2) the partial model of the operational structure, 3) the partial model of the organizational structure, 4) the partial model of the working persons involved in the development project and 5) the partial model of tools being used for product development. The partial model of the product to be developed is depicted in Figure 1 as an entity relationship model of different decomposition levels of a product and its relevant properties. The lowest level consists of so-called components that are interconnected by so-called interfaces. Interfaces between parts or subparts often need a close cooperation of the persons involved. Therefore, interfaces in our model can induce communication efforts.

Figure 2: Operational structure – properties of tasks and a task net with three design phases

The partial model of the operational structure defines tasks and orders of tasks (Figure 2). In computational terms, tasks can be considered as pointers to the work to be done, i.e. components to be transformed. Persons that are responsible for this transformation, adequate tools and resources etc., can be specified here. The most sensitive input variables are the expected workload (man hours) and the deadline of a task. By using the concept of deadlines it is no longer necessary to specify explicit predecessor-successor-relations. Nevertheless, it is also possible to define explicit predecessor-successor-relations. In doing so, milestones between design phases can be defined. In case of not fulfilling a defined target quality, a project leader can reallocate particular tasks to the respective person to increase quality. In order to proceed with a task of a subsequent phase all tasks of the previous phase have to be completed. Thus, the concept of so-called quality gates is realized in the simulation model. The mentioned definitions can be carried out in a so-called task net.

In the partial model of the organizational structure organizational relations and dependencies are defined. Furthermore, the organizational structure defines who is authorized and competent to use the right tools in the development process. An organizational unit can be either a position that is an organization of the lowest possible order, or an organization of higher order (Figure 3).

The most important partial model is the model of the person (Figure 4). The simulated persons are the driving force to initiate any action in the development project. We therefore
call our approach “person-driven” or “actor-driven”. In other words, the simulated person decides which task to process next and how to process it.

The model works like a queueing system, meaning a person has an “in-tray” with tasks to be processed. According to a bounded rational decision model (Licht et al. 2004, Schlick and Licht 2005) the person chooses the task to be processed next. Then, the person processes this task for a certain time. However, after some time the person reconsiders if, in between, another task has a higher priority. In that case the other task is processed next. When a task is finished it will be sent to a pool of “done tasks’.

To some extent, existing actor-oriented simulation models use similar mechanisms. However, when deciding between concurrent tasks, they apply FIFO or LIFO concepts or choose the tasks arbitrarily. Since in our model almost all tasks can be processed concurrently, such trivial concepts would not yield a realistic task execution sequence. Therefore, for when persons choose between concurrent tasks, we relied on bounded rational decision making theories (Ainslie and Monterosso 2004, Kahneman 2003, Steel and König 2006). If compared with economic approaches, our actors may be considered as “relatives” of the so-called “socialized homo oeconomicus”. This implies that decisions are based on goals and utility but also account for irrational measures like a temporal distance of rewards. For instance, a task with a close deadline will be assigned a high priority and a task with high importance for the company will also be assigned a high priority. Furthermore, if a person permanently processes a task, the priority of the task decreases over time. A person additionally assesses opportunity costs. Therefore, the person evaluates how much time to spend for getting familiar with a task. The simulated persons try to avoid opportunity costs and wait until the task is truly urgent.

There are several additional concepts being covered by our person models like communication skills, work experience, etc. For further details the reader may be referred to (Licht et al. 2004, Schlick and Licht 2005). Due to the bounded rational behavior of our simulated persons some of the above mentioned shortcomings of existing simulation models can be overcome:

- Realistic, semi-parallel or parallel task processing of multiple tasks is now possible.
- The simulation model now adequately models resource-induced task dependencies using individual efforts in terms of workload (man hours) for single tasks as input parameters of the simulation.

The model is capable of simulating two or more projects with parallel access to common resources. The model can also account for any other (e.g. irregular) work to be done – the additional tasks are simply put into the “in-tray” of the simulated person.

Last but not least, the simulated persons can use several tools for carrying out their development work. Tools are enabling resources for certain development tasks - for instance in detailed engineering - and have an impact on individual performance. With our tool partial model we modeled functional dependencies such as the influence on workload when processing a tool-supported task (Figure 5). In order to keep the model simple, we did not model all available software or hardware tools in product development, but only such tools that can be bottlenecks - for instance, expensive CAD software tools.

The five partial models described basically denote structures of product development processes. In the scope of integrating these partial models into an overall model single structural elements were linked in the sense of a flow model. The result is a process model in terms of system’s theory with inputs, outputs, elements with interdependent relations and transformation relations.

**MODEL IMPLEMENTATION AND VERIFICATION**

The conceptual model was transferred into a formal quantitative model. Therefore, the formalism of Timed Stochastic Colored Petri Nets was used. The Petri Net was verified with respect to boundedness, safety, blocking as well as conservation, and then was compiled into a computational model using standard Petri Net simulation software.

After implementation the model was verified. That is, by means of systematical experiments the consistency of the computational and the conceptual model were checked.

In order to be able to verify the simulation model a large number of simulation experiments on the basis of “toy problems” were conducted. Differently complex task nets were implemented and simulation runs with systematic parameter variations were computed.

The most trivial task net consisted of six tasks a person had to fulfill in parallel. A typical progression of this project is illustrated in Figure 6 as extended Gantt chart. Bars indicate

![Figure 4: Task Processing of the Simulated Persons](image)

![Figure 5: Partial Model of Tools](image)
that the person spent work efforts at that time. Frames from
the first beginning to the last end indicate the lead times
(durations) of tasks. It becomes obvious that these lead times
essentially consist of idle periods.

A simulated person can have preferences for different
working styles. The person can, for instance, prefer to
absolutely evenly perform parallel tasks, or alternatively, try
to first complete a task before working on the other tasks.
The person model in Figure 6 prefers a strategy in between
the two extremes. Once this person is acquainted with a task,
he or she tries to coherently continue the work – even if in
between another task was processed.

At the beginning of the project no task had a higher priority
than another. Task allocation by the person was therefore
done by chance, but the more unevenly the tasks were
processed, the more urgent the neglected tasks became. In
this case, the work strategy lead to a postponement of task 3.
Since the urgency of task 3 was quite high at its beginning
the task was completed very coherently.

In order to avoid confusion we would like to stress that our
model distinguishes between „importance“ and „priority“.
Importance is an input variable representing the value that
the person or the company assigns to the successful
completion of the task. A task to be done for a very
important customer may get a higher value of „importance“
than a minor customer development task. Conversely, the
priority that a person assigns to a task is a dependent
variable. Our simulated persons permanently compute the
priorities of each task - considering the importance, the
urgency as well as several other parameters that represent the
development goals. The simulation results on Figures 7 to 10
show this dependency.

In this case, two scenarios with different parameter settings
were simulated. In the first scenario a person had to process
two tasks with similar properties. The characteristics of the
degree of completion and of the urgency of the two tasks are
illustrated in Figures 7 and 8.

One can see that task 1 and task 2 are processed fairly
evenly, that is, when a person processes task 1 then the
urgency of task 2 increases, because the deadline comes
closer. This is a typical behavior of our socialized homo
oecomicus.

In the second scenario the same simulated person had to
process two almost identical tasks, although this time the
„importance“ of task 1 was much higher than the importance
of task 2. The characteristics of the degrees of completion

![Figure 6: One Person Performing Six Parallel Tasks](image1)

![Figure 7: Parallel Completion of Two Identical Tasks](image2)

![Figure 8: Respective Characteristics of Urgency u(t)](image3)

and of the urgency of the two tasks are illustrated in
Figures 9 and 10.

One can see clearly that – although the „urgency“ of task 2
permanently increases since the deadline comes closer and
the task has not been processed – task 1 will be processed
first. In this case, the priority of the task is dominated by
the importance and not by the urgency.

With respect to the management of multiple projects the
following scenario may illustrate how the simulation model
is working. The degree of abstraction of tasks in the
simulation model can be chosen arbitrarily. One could also
define an entire project as a task to be fulfilled by a person.
In the following scenario two persons are responsible for
completing three projects – each with identical work
volumes. Person 1 is responsible for projects A, B and C.
Person 2 is responsible for projects D, E and F. The strategy
of person 1 is to first complete a single project before
starting with the next one. Person 2 prefers to evenly work
on all three projects.

Simulation results of this scenario are depicted in Figure 11.
At a first glance both strategies lead to identical results for
finishing all three projects. However, when the projects are
processed evenly all projects need some more time units than
the sum of the sequentially processed projects. The reason
for this longer duration is the so-called setup time. Each time
a person changes attention to a new task he or she has to get
familiar with that task (or project) – even if that task has
been processed some time before. The longer it has been
since this task was last processed, the higher is the effort for
getting familiar with the task again. At this point a major benefit for project managers becomes obvious. Using this approach fine-grained modeling of the task structure is no longer necessary. Very long tasks (or even projects) can be processed in parallel. If, in the considered case, Person 2 was assigned an additional project or an important irregular task in between, the duration of all other projects would automatically increase.

**MODEL VALIDATION**

Up to this point we only studied whether the Petri Net simulation model is a suitable representation of the developed conceptual model. Now, the crucial question is: Is the simulation model a valid representation of a real complex product development project?

For this purpose project data in a company of the German electronics industry were collected. The company develops and manufactures electronic and mechatronical components and modules. The considered development project included tasks like conceptual design, designing and calculating connection diagrams, doing layout designs of circuit boards, etc. The collected data provided information about all five partial models. With respect to the efforts spent for each task the data of a barcode based labor time system could be accessed. This labor time system stores the real working hours of the product developers on specific tasks on a very fine-grained level (30 minutes accuracy).

The simulation model was parameterized with this data and simulation runs were conducted. The simulation results were reviewed by some experts of the company. They admitted that most of the tasks were simulated realistically.

The large amount of fine-grained empirical data of the development work processes allowed us to formulate the crucial null hypothesis for a project planner. This is “The total cycle time of the real development project and the total cycle time of the project simulation model do not significantly differ”. In order to test this hypothesis by means of inferential statistics, a sufficiently large number (n = 200) of simulation runs were computed and the output was analyzed. The normal distribution of the output data was validated with the Kolmogorov-Smirnov test. Finally, we were able to test the null hypothesis with Student’s t-test. The simulation results showed that the null hypothesis could not be rejected. That is, no significant differences between the real project duration and the simulated durations occurred. Moreover, a sensitivity analysis was carried out in order to study how the output changes if the main input parameters are varied.

**CONCLUSION AND IMPLICATIONS FOR MULTIPLE PROJECT MANAGEMENT**

The simulation model can help project managers in attaining a deeper understanding of their processes and how measures of interest would be affected if additional projects or tasks were assigned to particular team members, if deadlines were changed, or the work was structured differently.

For example, if a person works on five different projects and an additional project shall be assigned to that person then the project manager can assess how the tasks of the new project will probably be processed. Also – and this is a major benefit – he can assess how the other five projects will be affected by the new project – without changing any input parameter for these projects.

In order to plan multiple projects with the presented approach a project manager must first of all only define tasks with estimated efforts and deadlines and assign these tasks to persons. Efforts to be spent can be better estimated by project managers than task durations since durations depend on the course of concurrent tasks. Compared with existing approaches this enables a more intuitive modeling process. Fixed milestones can be defined in the task net. The simulation results consist of virtual courses of action for the projects. While exploring the results the project manager can
assess if the assigned deadlines can be met. In a trial and error fashion he or she can then change some input measures (deadlines, assignment of work to different persons, experience of particular persons, etc.) and assess the output measures of interest. For a more detailed analysis the project manager can model the underlying product structures in order to map information needs or bottleneck tools that could potentially influence the course of the simulated projects.

Alternatively, our model can be used to assess the schedule risk of an existing multiple project plan. All it takes are Gantt charts of the existing multiple projects (to identify the tasks and deadlines), the estimated task efforts, the persons to carry out the work and a first simulation experiment can be conducted. The simulation results can point the project manager to tasks that are unlikely to meet their deadlines.

Finally, our model can provide interesting insights in how the importance vs. urgency issue affects the course of projects. For project managers there are three major implications from our simulation studies:

1) Tasks that are halfheartedly assigned, that is, tasks without an explicit deadline – are very unlikely to be ever completed. This is not because persons are unmotivated or incapable though. Such behavior can be coherently explained with Picoeconomic (Ainslie and Monterosso 2004) or Temporal Motivation Theory (Steel and Köng 2006). If a project manager wants to make sure that a task gets priority it will not help to repeatedly emphasize the importance of that task – as long as there are concurrent tasks with a deadline they will get higher priority. Consequently the project manager should simply assign a (realistic) deadline.

2) Important tasks with an immense work load and therefore a long expected duration are likely to be neglected until they get urgent enough to be realized – usually the time left is too short for an excellent work result. To overcome this problem project managers should try to section long tasks into smaller portions and assign frequent deadlines.

3) Tasks with a very short deadline will be processed no matter how unimportant they are. A large amount of very short (and probably unimportant) tasks will always lead to persons assigning higher priorities to these short-term tasks and neglecting more important tasks that have a longer deadline. To overcome this problem project managers should more courageously avoid to assign such short term tasks to their project team – asking themselves if fulfilling the task is really worth neglecting the important long term task.

FUTURE RESEARCH

Ongoing research includes a model extension towards persons planning the task sequencing over a longer time horizon, developing a GUI for easy-to-use definition of simulation models and further validating the simulation model surveying new real project data.

ACKNOWLEDGEMENTS

This research was funded by the German Research Foundation (Deutsche Forschungsgemeinschaft), grant number Lu 373/26-1.

REFERENCES


PROJECT EVALUATION USING MONTE CARLO SIMULATION TECHNIQUE

Ahmad R. Sarfaraz
California State University, Northridge
18111 Nordhoff Street
Northridge, CA 91330
USA
Email: sarfaraz@csun.edu

KEYWORDS
Mutually exclusive alternatives, Monte Carlo simulation, engineering economy.

ABSTRACT
In evaluating of engineering projects, decision makers often confront situations where there are statistical distributions for economic parameters. As these projects become more complicated, derivations of the probability distribution in the economic measure of merit as a function of the unknown parameters can be tedious or impossible. This paper presents an application of the Monte Carlo simulation technique to analyze the economic consequences of some decision situations involving risk and uncertainty. Furthermore, it shows how Excel software can be used to solve such problems. A numerical example is also used to clarify the application of this useful technique.

INTRODUCTION
In general, engineering projects are evaluated by more than one feasible design alternatives. In searching feasible design alternatives, engineers usually consider several limitations including (1) lack of time and money, (2) perceptions of what will and what will not work, and (3) lack of knowledge. When the selection of one of these design alternatives excludes the other choices, the alternatives are called mutually exclusive alternatives. That is, the decision maker compares the alternatives and chooses the best to invest in.

The most straightforward technique for comparing mutually exclusive alternatives is to determine the equivalent worth of each alternative based on total investment at a discount rate or company’s minimum attractive rate of return (MARR). The equivalent worth method involves converting all cash flows to present worth (PW), or future worth (FW) using company’s minimum attractive rate of return (MARR). Then for investment alternatives, the one with the greatest positive equivalent worth is selected. Otherwise for the cost alternatives, the one with the least expensive is selected. This is done when the alternatives have equal lives. When the alternatives have different lives, the comparison must be made for equal-service periods. This is usually accomplished by performing the comparison over either a least common multiple or a specific study period. The annual worth (AW) method is often preferred to the PW or FW method, because the annual worth (AW) comparison is performed over the original alternative lives. This is a distinct advantage when comparing mutually exclusive alternatives with unequal lives. In making the above computations, it is assumed that all cash flows are to be known with certainty. That is the project engineer knows the values of the parameters with certainty.

LITERATURE REVIEW
In order to arrive at an acceptable solution for the project selection problem, a comprehensive understanding of the different applications of the Monte Carlo technique is essential. The significance that is attributed to the Monte Carlo applications is evident based on the sheer number of research papers that address this topic over the years. Monte Carlo simulation has been applied in the area of the facility location problem (Ridlehoover, 2004). The paper presented a methodology to help determine candidate locations and then used the Monte Carlo simulation technique to determine the ideal location of a new facility. The book published by (Young, 1996) uses the Monte Carlo simulation as a means of estimating technique to determine how much time and resource are required to carry out a piece of work to acceptable standards of performance. The Monte Carlo simulation has been applied in the area of diary science to estimate a distribution of breakeven prices, rather than a point estimate in a robotic milking system (Hyde and Engle, 2002). It is evident from the literature that the use of Monte Carlo simulation technique will evolve over time. Thus, this paper presents an application of the Monte Carlo simulation technique to analyze the economic consequences of some decision situations involving risk and uncertainty.

RISK AND UNCERTAINTY
One of the problems in selecting the best alternative among mutually exclusive alternatives is the effort required to estimate multiple parameters such as amount of required capital investment, estimated useful life, estimated annual incomes and expenses, interest rate, income tax rate, and project salvage value. Of these parameters, cost of new machinery and current tax rate
structure are the two parameters that can be well defined, but in several situations most of these parameters are unknown.

Two terms, uncertainty and risk, are used when an engineering project is faced by unknown economic parameters. Decisions under risk are decisions in which the project engineer selects the best alternative formulated in terms of assumed possible future outcomes whose probabilities can be estimated. Decisions under uncertainty, by contrast, are decisions that the project engineers cannot estimate the associated probabilities of the future outcomes. Although there is a technical distinction between risk and uncertainty, most project engineers use the terms interchangeably.

**MONTE CARLO SIMULATION TECHNIQUE**

There are a few of approaches for handling decisions under uncertainty and risk. Sensitivity analysis and spider plots are the two important approaches. Both approaches can provide insight into engineering projects but are not appropriate when there are statistical distributions for the economic parameters. Probability descriptions of the economic parameters allow further refinement of the analysis of economic risk and make the uncertainty associated with each alternative more explicit. As engineering projects become more complicated (when some of the economic factors of a project are uncertain), derivations of the probability distribution in the economic measure of merit as a function of the unknown parameters (random variables) can be tedious or impossible. Thus, a better technique such as Monte Carlo simulation technique is required to handle the problem. This paper provides solutions to such economy problems. Furthermore, it shows how Excel software can be used to solve these problems.

Monte Carlo simulation technique can be used as a practical method to analyze the economic consequences of some decision situations involving risk and uncertainty. This technique generates random outcomes for probabilistic parameters such as amount of required capital investment, useful life, annual incomes and expenses, interest rate, and project salvage value. The distribution parameters can be determined from data where they are available. The generated outcomes are then used to imitate the randomness inherent in the original problem.

**MODEL DEVELOPMENT**

Development of a model requires four steps. The first step is to construct analytical model based on the actual decision situation. The second step is to develop a probability distribution from subjective or historical data for each uncertain factor in the model. The probability distributions for the project parameters can be determined from data where they are available. The third step is to generate sample outcomes using the probability distribution for each uncertain factor. This process can be repeated for a large number of times so that a frequency distribution of trial outcomes can be established. One question is “how many simulation trials are necessary for an accurate approximation of the average outcome?” In general, the greater the number of simulation trials, the more accurate the approximation of the mean and standard deviation will be. Finally, the forth step is to use the resulting frequency distribution to make probabilistic statements for desired measures of merit.

**EXCEL AND MONTE CARLO SIMULATION TECHNIQUE**

Computer simulation packages, using Monte Carlo techniques, are readily available and are particularly well suited for sampling from various theoretical and data-defined statistical distributions. Of these computer programs, Excel provides a good application of the Monte Carlo simulation technique. In addition to engineering economy built-in factors in Excel, the Random Number Generation (RNG) can be used to generate random outcomes for probabilistic factors so as to imitate the randomness inherent in the original problem. The online help explains how to use the RNG to generate random numbers from a variety of probability distributions: normal, uniform, binomial, and Poisson. If any of the random variables is discrete, the discrete option can be used to generate the outcomes.

**NUMERICAL EXAMPLE**

The example shown in Table 1, illustrates the use of simulation to compare the investments of two alternative projects based on their equivalent annual worth. We will let annual operating expense (AOC), salvage value (SV), and useful live (n) are the three parameters that are unknown. We are interested in developing a simulation of 50 sample points of annual worth (AW) using the probability distribution specified for each random variable. We assume MARR is 12% per year.

<table>
<thead>
<tr>
<th>Alternative</th>
<th>A</th>
<th>B</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Annual Operating Cost (AOC)</strong></td>
<td>Normal Distribution: Mean: 8,000 Std Dev.:1,000</td>
<td>Normal Distribution: Mean: 3,000 Std Dev.: 500</td>
</tr>
<tr>
<td><strong>Salvage Value (SV)</strong></td>
<td>Uniform Distribution: 0 to 1,000</td>
<td>Uniform Distribution: 500 to 2,000</td>
</tr>
<tr>
<td><strong>Useful Life, (n)</strong></td>
<td>Discrete uniform 3 to 8 years with equal probability</td>
<td>Discrete uniform 3 to 7 years with equal probability</td>
</tr>
</tbody>
</table>

Table 1: Numerical Data for Both Alternatives A and B
To use Monte Carlo simulation technique, it is necessary to generate a number of trails for each of the parameters based on the given conditions for each alternative. Basically, the results of any simulation study are subject to the number of times the simulation occurred. Thus, the more simulation trails for which the simulation is conducted, the more accurate the result we get. To show the application of the technique in evaluating engineering projects, 50 trials for each of the parameters were simulated in this paper.

The simulated outcome can be generated in Excel by selecting “Tool” then “Data Analysis” and “Random Number Generation”. It is furthered assumed that the required capital investment for both alternatives are the same. Once all simulated outcomes are generated it is necessary to compute the Net Annual Operating Costs (NAOC) for each of the alternative. Since the alternatives have different lives, the comparison was made using the AW method. This is accomplished by taking the AOC and subtracting the equivalent annual worth of salvage value. Using engineering economy A/F factor (finding uniform amount given future value, or simply sinking fund factor), this can be expressed as follows.

\[ \text{NAOC} = \text{AOC} + \text{SV} \times (A/F, 12\%, n) \]  

<table>
<thead>
<tr>
<th>Alternative</th>
<th>A</th>
<th>B</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Annual Operating Cost (AOC)</strong></td>
<td>Normal Distribution: Mean: 8,000 Standard Deviation: 1,000</td>
<td>Normal Distribution: Mean: 3,000 Standard Deviation: 500</td>
</tr>
<tr>
<td><strong>Salvage Value (SV)</strong></td>
<td>Uniform Distribution 0 to 1,000</td>
<td>Uniform Distribution 500 to 2,000</td>
</tr>
<tr>
<td><strong>Useful Life, (n)</strong></td>
<td>Discrete uniform 3 to 8 years with equal probability</td>
<td>Discrete uniform 3 to 7 years with equal probability</td>
</tr>
</tbody>
</table>

The “pmt” function in Excel was used to carry out these calculations. The average NAOC for the entire alternative is then calculated by taking the total average of each of the 50 trials. The net results of the calculated average NAOC for both alternatives are shown in Table 2. Based on the calculations, alternative B would be the best to choose. This is because the average NAOC is the least expensive alternative.

<table>
<thead>
<tr>
<th>Alternative</th>
<th>Net Annual Operating Costs (NAOC), $</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>7,787</td>
</tr>
<tr>
<td>B</td>
<td>2,731</td>
</tr>
</tbody>
</table>

**VERIFICATION OF THE SIMULATION RESULT**

It is often difficult to validate the results of a simulation model. However, simulation most often is used whenever there is no other technique to solve the problem. In general, the results of a simulation model are statistical measures that can be further analyzed to determine their degree of accuracy. One of the most frequently used tools for this analysis is confidence limits. For example, a 95% confidence interval can be constructed for the AW results performed by the Monte Carlo simulation technique. Although we cannot be certain that the AW of each alternative will exactly equal the population mean, we can be 95% confident that the true value of the AW will be between the upper confidence limit and lower confidence limit.

**CONCLUSION**

The use of Excel with Monte Carlo simulation technique evaluates engineering projects problem solutions more realistically. Probability descriptions of input variables and Monte Carlo sampling together provide a practical method of finding the distribution of the desired output given the various random and deterministic input variables. The results of such analyses give better information for making decisions.

**REFERENCES**


**AUTHOR BIOGRAPHY**

**AHMAD R. SARFARAZ** obtained his Ph. D. degree from West Virginia University in 1990. Since then, he has been on the faculty of Abadan Institute of Technology and Amirkabir University of Technology, and now at the California State University, Northridge where he is Assistant Professor in the Department of Manufacturing Systems Engineering and Management. In addition to teaching courses on Facilities Planning and Design, he has taught Engineering Economy and Engineering Project Management at the undergraduate level and Advanced Engineering Management, Engineering Economic Analysis, and Engineering Statistics at the graduate level.
SIMULATION DRIVEN DESIGN SYSTEMS
Architecture for integrated simulation driven design systems

Kaushik Sinha, Roland Haas
DaimlerChrysler Research and Technology India, Bangalore
E-Mail: kaushik.sinha@daimlerchrysler.com; Roland.H.Haas@daimlerchrysler.com

ABSTRACT

This paper presents an approach that makes numerical simulation and optimization an integral part of the product development life cycle wherein mathematical modeling, simulation and optimization form the basis of decision making at every step in the product development cycle. Here, we present a process architecture that tightly couples digital simulation and multi-disciplinary optimization with the product development cycles and helps in design innovations. This method integrates CAD, CAE and numerical optimization inside a single system resulting in “first time right” and manufacturability of design, thereby reducing the cycle time significantly. This also ensures optimized product variants aimed at meeting specific set of customer requirements as an outcome of the integrated development process.

KEYWORDS: innovation, design, multi-objective optimization, multi-disciplinary optimization, CAD, CAE, Optimization integration, process architecture.

Introduction:

In today’s highly competitive business environment, large-scale engineering (LSE) organizations are constantly searching ways to differentiate their products from their competitors, increasing the level of innovation and packing additional features and functionality in their products to reduce the development cycle time. This calls for a well-integrated product development cycle which ensures innovation in the process and encourages designs that are “first time right” and thereby reducing the number of prototypes. PDM systems on their own do not ensure innovative designs and also correctness of design.

In typical product development processes, there is a sequence of events evolving ideas and concepts into conceptual design and/or architecture, then preliminary design and finally detailed design and manufacturing. This process is generally divided into three phases as shown in Figure 1. The paradox facing the designer lies in the problem that in order to gain knowledge about a particular system architecture or design, he must first make decisions regarding the architecture or design of the system or product under development. These decisions result in a loss of design freedom. Once enough information has been gained, for example through testing, it may be too late to act upon it appropriately. The International Council on Systems Engineering (INCOSE) estimates that 70%-90% of the development cost of complex systems is pre-determined after only 5%-10% of the development time has been completed.

Multidisciplinary system architecture modeling and analysis allows the system architect or designer to gain more knowledge about the system behavior while at the same time retaining design freedom. This approach then has the potential to break the paradox described earlier. Figure 1 shows how the multidisciplinary system architecture modelling methodology affects system knowledge and design freedom during the product development process.

Fig. 1: Re-engineered design process to gain early knowledge early and retain design freedom

In essence, multidisciplinary system architecture modeling allows the designer to gain valuable knowledge regarding the potential performance of possible system architectures and/or designs. Without being forced to make design decisions too early, the designer maintains design freedom longer into the system architecting and/or design process. In the context of new technology infusion into existing systems, an additional benefit is the ability to computationally assess the potential cost/performance trade-off before committing to invest significant R&D budgets, since most of the project cost is expended in the preliminary and detailed design stages.

In this paper, we present a process architecture that tightly couples digital simulation and multi-disciplinary
optimization with the product development cycles and helps in design innovations. This method integrates CAD, CAE and numerical optimization inside a single system resulting in “first time right” and manufacturability of design, thereby reducing the cycle time significantly. This also ensures optimized product variants aimed at meeting specific set of customer requirements as an outcome of the integrated development process.

**Process flow and Architecture:**

A typical product data management (PDM) system controls product related data and organizes the workflow through the product development life cycle. The two major aspects that PDM systems focus on is:

1. Centralized and structured storage of product data
2. Workflow management.

On the whole, PDM system presents an organization with a mechanism for improving communication and cooperation among diverse groups involved in the product development process. IMDOS, combined with existing PDM systems, forms a simulation-driven product design system. All product performance related information is centrally stored within the product development framework. The IMDOS architecture uses basic product related data from PDM systems, integrates performance validation, optimal product synthesis and decision making in a multi-disciplinary framework.

**Basic components of IMDOS framework:**

![Diagram showing IMDOS modules and basic flow](image)

Fig. 2: Proposed IMDOS modules and basic flow

The basic template component contains all potential design variables and the parametric design rules. This information is passed to the parametric CAD system. The green box demarcates the IMDOS from prevailing PDM systems within which parametric CAD system is an essential component. IMDOS requires information from CAD system, basic vehicle definition template and corporate materials database as shown in Fig. 2.

Core IMDOS provides detailed simulation-driven product synthesis and validation features. It enables bidirectional data transfer with CAD system, since it is possible to put back synthesized vehicle data into the CAD system through common vehicle database.

**Common Vehicle Database:**

The common vehicle database acts as the gateway for information exchange between existing PDM system and IMDOS. This system assumes a parametric format that for a specific instantiation (that is, for a specific generated configuration) of the parameters defined in the basic template will generate a common vehicle description.

The Common Vehicle Database contains materials data as well as mass and inertia characteristics of components in addition to mandatory geometric information (CAD data). Design variables or parameters for each discipline are partitioned into global and local set of variables. Each discipline should share a common description for all global parameters or design variables. This is a basic requirement of IMDOS system.

The common vehicle database stores all the numerically computed performance measures. Therefore only results database updates it with the computed results.

**Results Database:**

This is the heart of IMDOS. Design of results database is very crucial to proper functioning of IMDOS. Results database holds details of type of data and their inter-relationships or mapping. This database is solely responsible to communicate with the common vehicle database that acts as a gateway to existing PDM system. Each analysis discipline (e.g. Crash, NVH, CFD, etc) supplies some specific subsets of their output data to the results database. Each instance of discipline specific analysis also put-in additional details (meta-data) into results database.

![Relational description within results database](image)

Fig. 3: Relational description within results database -- snapshot

For a given response identifier, the results database identifies all designs residing within the database and return the specific response value to the model generator module (as indicated in fig. 3). This database holds a pointer to explicit mathematical model for any discipline, if it exists. This model is made
available for use in design studies. Results database contain all potential design variables that are present in the basic template and store a historical link-relation table amongst potential and actual (reduced) set of design variables, if any. User then is able to choose which of the available variables are to be included in the meta-model development process. To further enhance the functionality and reusability, the results database stores a historical link-relation table between potential and actual (reduced) set of design variables, if there are any. This database, along with meta-model generation module, is used to build explicit response surfaces for upfront preliminary design in a vehicle development program.

Results database also acts as an enabler for multi-disciplinary, multi-objective optimization within an integrated environment, making continuous product innovation and alternate concept generation possible.

Meta-model generation and design optimization module:

This is the performance validation and design synthesis component within IMDOS framework. The primary function for this block is to generate response surfaces for multitude of responses, measuring product performance. These relations are subsequently used for design optimization, product reliability assessment, etc.

![Data flow within meta-model generation module](image)

Fig. 4: Data flow within meta-model generation module

This module extracts data from results database for model building and also sends a request if more sampling points are required. Results database, in turn processes the same through common vehicle database to generate additional vehicle description (refer to figure 4). The meta-model generation module is primarily responsible for two functions:

1. Variable screening to filter a subset of design variable from a bigger list of potential variables, thereby reducing the problem size.
2. Build a predictive response surface model with validation.

Case study:

To illustrate the use of the IMDOS framework, an example of a vehicle design study for crashworthiness under various impact and passenger safety criteria is presented.

Please note that the values indicated do not represent any real-life vehicle data.

A full vehicle is to be designed/optimized under multiple criteria based on full frontal impact, 50% frontal impact, side impact and roof crush scenario.

The identified design variables are segregated into a set of global or common design variables (table 1) and discipline specific local design variables (tables 2 through 4). For each kind of scenario, the responses are provided in table 5.

There are 29 potential design variables and forms a part of the basic template. They depict a vehicle description, which is stored in the common vehicle database.

The design variables are partitioned and each discipline shares a common description for all the variables using unique name.

<table>
<thead>
<tr>
<th>No</th>
<th>Design variable name</th>
<th>Lower bound (mm)</th>
<th>Upper bound (mm)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Windshield</td>
<td>2.6</td>
<td>5.0</td>
</tr>
<tr>
<td>2</td>
<td>Roof panel</td>
<td>0.6</td>
<td>1.5</td>
</tr>
<tr>
<td>3</td>
<td>Roof rail</td>
<td>0.6</td>
<td>1.5</td>
</tr>
<tr>
<td>4</td>
<td>Front roof cross member</td>
<td>0.6</td>
<td>1.5</td>
</tr>
<tr>
<td>5</td>
<td>Rear roof cross member</td>
<td>0.6</td>
<td>1.5</td>
</tr>
<tr>
<td>6</td>
<td>A-pillar</td>
<td>0.6</td>
<td>1.5</td>
</tr>
<tr>
<td>7</td>
<td>B-pillar 1</td>
<td>1.0</td>
<td>1.8</td>
</tr>
<tr>
<td>8</td>
<td>B-pillar 2</td>
<td>1.0</td>
<td>1.8</td>
</tr>
<tr>
<td>9</td>
<td>B-pillar 3</td>
<td>1.2</td>
<td>2.0</td>
</tr>
<tr>
<td>10</td>
<td>C-pillar</td>
<td>0.8</td>
<td>1.8</td>
</tr>
</tbody>
</table>

Table 1. – Common design variable set with design limits

<table>
<thead>
<tr>
<th>No</th>
<th>Design variable name</th>
<th>Lower limit (mm)</th>
<th>Upper limit (mm)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Rail 1</td>
<td>1.0</td>
<td>3.0</td>
</tr>
<tr>
<td>2</td>
<td>Rail 2</td>
<td>1.0</td>
<td>3.0</td>
</tr>
<tr>
<td>3</td>
<td>Rail 3</td>
<td>1.0</td>
<td>3.0</td>
</tr>
<tr>
<td>4</td>
<td>Rail 4</td>
<td>1.0</td>
<td>3.0</td>
</tr>
</tbody>
</table>

Table 2. – Local design variable set with design limits for frontal and 50% offset impact

<table>
<thead>
<tr>
<th>No</th>
<th>Design variable name</th>
<th>Lower limit (mm)</th>
<th>Upper limit (mm)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Door reinforcement</td>
<td>1.2</td>
<td>2.0</td>
</tr>
<tr>
<td>2</td>
<td>Rocker outer</td>
<td>0.7</td>
<td>1.5</td>
</tr>
<tr>
<td>3</td>
<td>Rocker inner</td>
<td>1.2</td>
<td>2.3</td>
</tr>
<tr>
<td>4</td>
<td>Cross member</td>
<td>1.5</td>
<td>3.0</td>
</tr>
<tr>
<td>5</td>
<td>B-pillar reinforcement</td>
<td>1.5</td>
<td>3.0</td>
</tr>
</tbody>
</table>

Table 3. – Local design variable set for Side impact (5)
Table 3. – Local design variable set with design limits for side impact

<table>
<thead>
<tr>
<th>No</th>
<th>Design variable name</th>
<th>Lower limit</th>
<th>Upper limit</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Front door</td>
<td>0.4</td>
<td>1.0</td>
</tr>
<tr>
<td>2</td>
<td>Front door inner</td>
<td>0.5</td>
<td>1.2</td>
</tr>
<tr>
<td>3</td>
<td>Rear door</td>
<td>0.7</td>
<td>1.5</td>
</tr>
<tr>
<td>4</td>
<td>Mat A-pillar 1 (Gpa)</td>
<td>0.192</td>
<td>0.345</td>
</tr>
<tr>
<td>5</td>
<td>Mat A-pillar 2</td>
<td>0.192</td>
<td>0.345</td>
</tr>
<tr>
<td>6</td>
<td>Mat A-pillar 3</td>
<td>0.192</td>
<td>0.345</td>
</tr>
<tr>
<td>7</td>
<td>Mat B-pillar 1</td>
<td>0.192</td>
<td>0.345</td>
</tr>
<tr>
<td>8</td>
<td>Mat B-pillar 2</td>
<td>0.192</td>
<td>0.345</td>
</tr>
<tr>
<td>9</td>
<td>Mat B-pillar 3</td>
<td>0.192</td>
<td>0.345</td>
</tr>
<tr>
<td>10</td>
<td>Mat front door inner 2</td>
<td>0.192</td>
<td>0.345</td>
</tr>
</tbody>
</table>

Table 4. – Local design variable set with design limits for roof crush

<table>
<thead>
<tr>
<th>Analysis type</th>
<th>Response identifier</th>
</tr>
</thead>
<tbody>
<tr>
<td>Frontal impact</td>
<td>Total vehicle crush (mm)</td>
</tr>
<tr>
<td></td>
<td>Maximum rail force (kN)</td>
</tr>
<tr>
<td></td>
<td>Dash intrusion (mm)</td>
</tr>
<tr>
<td></td>
<td>HIC</td>
</tr>
<tr>
<td></td>
<td>Chest G (g)</td>
</tr>
<tr>
<td>50% offset frontal impact</td>
<td>Intrusion footrest (mm)</td>
</tr>
<tr>
<td></td>
<td>Intrusion toe pan left (mm)</td>
</tr>
<tr>
<td></td>
<td>Intrusion toe pan centre (mm)</td>
</tr>
<tr>
<td></td>
<td>Intrusion toe pan right (mm)</td>
</tr>
<tr>
<td>Side impact</td>
<td>B-pillar displacement (mm)</td>
</tr>
<tr>
<td></td>
<td>Door displacement (mm)</td>
</tr>
<tr>
<td></td>
<td>Rib displacement 1 (mm)</td>
</tr>
<tr>
<td></td>
<td>Rib displacement 2 (mm)</td>
</tr>
<tr>
<td></td>
<td>Rib displacement 3 (mm)</td>
</tr>
<tr>
<td></td>
<td>V°C 1</td>
</tr>
<tr>
<td></td>
<td>V°C 2</td>
</tr>
<tr>
<td></td>
<td>V°C 3</td>
</tr>
<tr>
<td></td>
<td>Pubic symphysis force (kN)</td>
</tr>
<tr>
<td>Roof crush</td>
<td>Resistant force (kN)</td>
</tr>
<tr>
<td></td>
<td>Crush distance (mm)</td>
</tr>
</tbody>
</table>

Table 5. – Discipline specific responses

**Note: Another independent or common response quantity calculated is Structural weight of the vehicle.**

These responses outlined in table 5 above are stored in the results database along with associated meta-data like parameter description (figure 4), response description, etc.

In this example, there is no historical link-relation table amongst design variables and the results database contains all 29 potential design variables. The response surfaces for each of the listed response quantities are built inside the meta-model generation module. Polynomial predictive models (piecewise-polynomial) are built on the reduced, significant set of design variables. In order to accomplish this function, meta-model generation module communicates with results database to get the response values corresponding to each row in DOE (Design of Experiments) table. These response surface models are then accessed by the multi-objective design optimization code GDOT (Generic Design Optimization Toolkit, DCRTI proprietary) that outputs the pareto optimal front. The pareto optimal data is analyzed and the decision making block selects a subset of points on the optimal pareto front as preferred design points. In this case, this is performed using a set of rules. User can perform this job interactively as well. This information about the chosen set of designs is sent to results database, which gets updated with the corresponding set of results and parameter values.

**Conclusion:**

This paper introduced and described the proposed IMDOS framework that works on top of existing PDM system and integrates product performance validation and synthesis. This system consists of functionally different modules that tightly couple numerical simulation and design optimization to the product development process in addition to the functionalities already offered by the PDM systems. IMDOS framework is a step towards realizing a simulation-driven product development process and thereby reducing the cycle time for product development and enabling product innovation. This, in effect, reduces the number of design changes and helps to increase design knowledge early in the development phase (refer to fig. 4 and 5). The reduction in product development time is achieved by inherent validation and synthesis of the product leading to “first time right” design.

**References:**

Ph.D. Dissertation, Woodruff School of Mechanical Engineering, Georgia Institute of Technology, Atlanta, Georgia, 1995.
LIFE-CYCLE FEATURES FOR MECHANICAL PRODUCTS ANALYSIS AND SIMULATION

Christian Mascele
Ecole Polytechnique de Montréal, Département de Génie Mécanique
C.P. 6079, Succ. Centre-Ville, Montréal, Québec, H3C 3A7 Canada
E-mail: christian.mascele@polymtl.ca

KEYWORDS
Simulation, Modeling for X; Product Features; Life Cycle.

ABSTRACT

This paper deals with a representation of life-cycle features. Modeling the product at each stage of the life-cycle process, simulation processes and taking into account all pertinent geometrical, technological and functional data, distinguish our model from ones described in the literature. A representation such as this also greatly contributes to the designing of a system that includes the various stages in the development and manufacture of a product. To reach our goal, we let the user interactively input a set of features that cannot be supported by a solid modeler. The system automatically deduces other features.

INTRODUCTION

The implementation of concurrent engineering in the design process has contributed to lower the development time and manufacturing cost of products. But there is still a great lack in integrating the various life cycle aspects of the electromechanical products to the design process. This is due to several reasons among which: (1) the traditional approach of the design is centered only on the functionality of the product, (2) the lack of a design tool that offers a feed back to the designer on the impact of his design decision on the assembly, recycle, environment and sustainability. Given the object-oriented structure that we chose, the different features can be abstractly represented as object attributes. This can assist the engineers in the design of a new product, but also in the task and resources planning. Examples are given to validate the concept. The studies of the Design For X (DFX, e.g., manufacturing, (dis)assembly, sustainability, and environment), which deals with the various steps of the life cycle such as assembly, disassembly, environment, recycle and sustainability continue to receive great attention by the manufacturing and design community. To date, much work has been achieved on various aspects of the design for assembly, disassembly sequences, features recognition, design for environment and the integration with Computer Aided Design (CAD). This makes it possible to incorporate effectively some aspects of the downstream activities knowledge into the preliminary design process. However, most modern CAD systems do not directly permit direct integration of the design features in order to discover the impact of the design decisions on the assembly, environment, recycling etc. In this research, we effectively integrate the downstream activities in the early process of the design; the downstream activities are the design for X activities which consist of the design for assembly, design for disassembly, design for recycle, and design for sustainability. The design requirements are considered to be the generalized product features and listed as the different design for X parameters. The generalized product features are then integrated in a CAD model of the product. Since the regular CAD software lack the ability to represent the design features, an object oriented design modeling software program namely ACIS is used to model the product in order to integrate the different design features previously defined. The design features form the product features database and a knowledge-based approach is developed to integrate the database of the design features and support the DFX design decisions. Using the database features created of the product model, the knowledge-based approach and the object oriented model of the product, the designer can foresee the impact of his decisions on the assembly, the environment and the recycling. The design methodology is implemented in a physical model to evaluate its practicability.

Generally, a hierarchical product structure has been proposed because of the natural decomposition process used by the designer during the top-down design process. An alternate decision-based representation to support design exploration and optimization during top-down design has been proposed by Kim and Szykman (Kim and Szykman 1995) with an underlying hierarchical product structure. Henson et al. (Henson et al 1993) also related the functional specifications to the physical structure of the product using a hierarchical structure. Many representations have focused on capturing the mating constraints between components. Various detailed assembly representations have evolved including kinematics-based (Rajan and Nof 1996, Turner and Sabramanian 1992), geometry-based (Anantha et al. 1996, Ngoi and Tan 1996, Wang and Ozsoy 1993) and feature-based (Delchambre 1992, Jabbour et al 1993, Van Holland and Bronsvoort 1996). Pratt (Pratt 1996) has developed a more generic geometric constraint representation for use in describing component as well as assembly constraints. The feature-based assembly representations attempt to provide a higher-level assembly model which includes various types of assembly features. However, the survey of the literature has revealed a non-unique definition of these features. The modeling approach that analyzes liaisons as the relations
between components is also inadequate to support other activities related to the modeling approach, especially the dimensioning of the product (Ngoc and Tan 1996, Wang and Ozsoy 1993) requiring that all contacts between mating features on parts be analyzed. Recently, environment concerns, recycling and design for disassembly became an emphasis in the manufacturing community mainly because of the huge quantity of used product that is being discarded each year (Ishii et al. 1994, Ishii and Eubanks 1993, Rose 2000, Qian et al. 2001, Mosovsky et al. 2001, Ishii et al. 2003, Khan et al. 2004, Fiksel 2003). It has been recognized that disassembly which may be considered as a reverse of the assembly is necessary to be considered in the design process in order to make recycling more viable. In general, features correspond to the information in relation to the concepts of interest in an application. In the case of the mechanical design, features can relate to the material, the energy and other design information like design intent, life cycle strategies, alternatives etc. Most researches have been concentrated on the form features. By using a feature interface to a geometric modeler, the design engineer can be informed by possible shortcomings regarding a predefined life cycle application automatically rather than having to analyze the design manually. The geometric model is constructed using primitives in the form of slots, pockets and so on from a form feature library. Geometric Boolean operations link the features to the geometric part. The system stores the information about the feature locations and parameters in the database. A knowledge base expert system may help retrieve the information from the database to analyze a given part design. Product are not constituted by form features only, other useful information regarding the life cycle applications, design intent and alternatives need to be incorporated as well in order to make the designer aware of the consequence of his decisions during the design process. From this literature review, it can easily be concluded that the modeling approach remains essentially the same from one author to another: it consists of characteristics related to the components of a product in its assembled state. This type of approach does not take into consideration the effects that a modification in the state will have on the various characteristics defined for the product model. The above facts emphasize the importance of modeling the product at each stage in the life-cycle process. They cannot be explicitly emphasized without making the life-cycle states an integral part of the product model.

CAPABILITIES AND FUNCTIONS OF THE PLFS

Given the limits presented by the models cited, a system called PLFS (Product Life-cycle Feature System) is proposed to support the various activities involved during the product life. Because of its more comprehensive approach, this system both bridges the gap that exists in current CAD systems and will prove valuable in the design of all-inclusive systems combining the various stages in the development of a product. It also differs from the studies mentioned above in two important aspects. First, the system is designed to let the user infer a complete set of technological and functional features that are not explicitly present in the geometric representation stemming from the solid modeler, but that are useful in supporting the product model. The relationships among components can be automatically clarified by using a hierarchical procedure that captures the relationships between local regions of mating parts at the first level. Second, it allows for product modeling at each step in the X process (DFX). The database structure defined in the PLFS system is based on a sequence structure, which allows the user to integrate states into the product model and allows the system to automatically deduce features relative to each state by comparing its structure to that of the assembled state. Finally, the object-oriented structure used in the system also allows the user to save and regenerate the structures of the various states. This allows, firstly, the application a posteriori of evaluation criteria in the design for X and, secondly, ensures an interface with a module that generates X resources in the case where this task is also defined a posteriori. To meet the objectives of using the proposed model, this study examined the various X stages and, for each stage, determined the type of features that must be generated by the system to assist the engineer. As mentioned earlier, non geometric features that are not directly supported by the solid modeler are added during the product modeling stage. Figure 1 illustrates the role of PLFS as a support of the product modeling during the analysis of its life-cycle. Such a role is provided through different functionalities which also lead to the generation of a multi-level product model based on the definition of models for different types of entities. As illustrated on Figure 1 these functionalities are related to three major tasks. The definition of the product hierarchical structure which integrates the models of the components specified in their designated locations in the assembled state. The recognition of component features pertaining to each state of the product. The system automatically deduces the features relative to an intermediate state by comparing its structure to the one of the life-cycle state. To meet the objectives of this study, a hierarchical structure was first created.

PROPOSED LIFE-CYCLE FEATURES

We first created an assembly structure that contained the geometric and non-geometric features of individuals parts specified in their designated locations in the assembly. The system adopts a B-Rep (Boundary Representation) model, which describes an object using all its faces, therefore creating the envelope, each face being limited by an edge loop. During the modeling stage, the user may also specify a set of non geometric data by means of four interactive modules that can be executed separately. Liaisons are important features in a product because they are used to infer the assembly/disassembly directions of parts and other features mentioned above that play a major role in the generation and evaluation of operation plans and product states or impacts.
As mentioned in the introduction, in the PLFS system it is assumed that assembly/disassembly operations consist of two stages. The first stage is referred to as the prepositioning stage and the second, the process stage, to differentiate between the state of parts before and after the application of a process. The syntax used for the calculations is shown in Figure 2. The process of volume generation does not take into consideration the faces that are at the origin of the existence of a sub-face characterized by an assembly/disassembly process that contributes to a deformation (such as crimping, clipsage, etc.). This same rule applies equally to each face declared to undergo a processing equivalent to an assembly/disassembly process of the same type as those mentioned above. Figure 3 also shows the necessity of using a hierarchical multi-level structure. One can set aside the solid representation for part and subassembly levels, resulting in a greater portability of applications, life-cycle features and a reasonable processing time, even for complex assemblies. In the PLFS system, processes are divided into two categories, according to the technique involved. Processes leading to the creation of liaisons between parts and processes which cause a face to undergo some kind of processing, such as manufacturing, painting, rusting, etc. Figure 4 illustrates the solid model of an assembled product as it was created and represented during the modeling stage.

\[
P_j = \bigcup_{l=1}^{1} \bigcup_{m,j=1}^{1} LIf(l, m, j)
\]

\[
SA_k = \bigcup_{l=1}^{1} \bigcup_{m,j=1}^{1} LIf(l, m, j)
\]

\[
SE_{pj} = \bigcup_{l=1}^{1} \bigcup_{m,j=1}^{1} LIf(l, m, j)
\]

**Figure 1:** Role of PLFS as a support of the product modeling

**Figure 2:** Syntax used for the liaison calculations

**Figure 3:** Hierarchical multi-level structure
REPRESENTATION OF THE LIFE-CYCLE STATES

To clearly illustrate the capability of the PLFS system to model a product at each life-cycle stage, an interactive module was developed to allow the simulation of each stage, both for assembly and disassembly, with a procedure that automatically generates a new configuration of the product representing its state, according to user stipulated specifications related to each operation. These specifications concern the components, and the direction of assembly or disassembly of each component. On the basis of the previous stage, the user may select any state of the product, schematize its structure (Figure 5) and examine the various features relative to that state.

![Figure 4: Solid model of a micro-engine](image)

The system then checks for a possible collision between the gripper and other parts of the product. It can also examine the feasibility of a process such as disassembly, maintenance, welding, gluing, etc. The user can display the free parts of the faces (or free sub-faces) selected for grasping. For example, with respect to state (11), the system deduced on the piston subassembly the free sub-faces shown darkened Figure 6. The configuration of the product may understand as a set of entities, each of them working separately. But these entities represent, on the first hand, the subset of assembled components and, secondly, the disassembled components. Therefore, we defined a product’s state as the set of entities that are considered separately at any step of the life-cycle stages. For example, with the product shown in Figure 7, we associate the Product State Graph (PSG) shown in Figure 8 representing the product’s disassembly sequences.

![Figure 5: Intermediate state](image)

Figure 6: Free sub-face for state 11

![Figure 7: Intermediate state](image)

MODEL STRUCTURE

A developed computer program was used to validate the proposed model structure on the basis of the prototype. It uses the ACIS solid modeler, which was developed and marketed by Spatial Technology, Inc. It is produced as a library of ACIS modeler procedures written in the C++ language and is equipped with a graphic interface, which allows solid objects to be defined, handled and displayed. In order to create his own application, the user can use standard commands interpreted develop a C++ code that accesses the ACIS library directly. The user can also extend ACIS, integrating other classes of objects that are not included in its node and creating customized commands. This last step also corresponds to the methodology supporting the PLFS system for the modeling of a product. Its advantages include: allowing the display of parts and their various geometric features (faces, sub-faces, etc.). Working through interactive modules to add the various geometric features and to infer the features relative to each state. Operating certain procedures developed in ACIS, such as saving data as a text file that can always be retrieved to rebuild the database structure. As shown in Figure 3, each subassembly of the product, or the product itself, may be handled using the PLFS system. Therefore, a model is built to allow for a simulation of the main product’s assembly/disassembly process from the final assembly/disassembly system to the sub-lines.

![Figure 8: Product State Graph (PSG)](image)

CONCLUSION

The life-cycle product model presented in this paper supports the design of a mechanical product for X, the automatic or aided determination of its sequences and the resources required for its maintenance, assembly or disassembly, recycling, etc. The results obtained were encouraging. For the micro-engine tested (made of 15 parts involving 668 faces), it took the user approximately 40 minutes to interactively associate various non geometric features with the parts and their faces and identify the subassemblies. To model liaisons between components of the assembled state of a product, the system CPU took 112
seconds to automatically detect 36 contacts between faces, generate contact sub-faces, and model their liaisons. And it took only seconds or even a fraction of a second to deduce the features relative to the intermediate states. The results clearly prove that we met the objective defined for the PLFS system.

Figure 7: Micro-engine assembly drawing

Figure 8: State graph of the micro-engine in assembly

REFERENCES


Van Holland, W. and Bronsvoot, 1996, “Assembly features and sequence planning”, Proceedings of the 5th IFIP Workshop on Geometric Modeling in CAD.


BIOGRAPHY

CHRISTIAN MASCLE was born in Nyons, France and he is a full professor at Department of Mechanical Engineering, École Polytechnique de Montréal. He received his PhD in Microtechnic Eng. from École polytechnique fédérale of Lausanne, his BSc degree in Mechanical Eng. from École Polytechnique de Montréal. His research interests include assembly/disassembly and task planning, product life-cycle modeling, intelligent CAD and engineering applications of object oriented programming.
INDUSTRIAL
PROCESS
ANALYSIS
SIMULATION
SIMULATION CASE STUDY: AN ANALYSIS OF REACTOR DEFUELING LEAD TIME IN A POWER PLANT

Habtom Mebrahtu
Caroline Strange
Rob Walker
Faculty of Science & Technology
Anglia Ruskin University
Victoria Road South, Chelmsford
Essex CM1 1LL, United Kingdom
H.Mebrahtu@anglia.ac.uk
r.w.walker@anglia.ac.uk

Stefan Roman
Magnox Electric Ltd.
Bradwell Power Station
Bradwell-on-Sea
Southminster, Essex, CM0 7HP
stefan.roman@magnox.co.uk

KEYWORDS
Simulation, Optimisation, Resource Smoothing

ABSTRACT
Simulation has been used for many years to increase understanding of a process and enable that process to be optimised. In this case study simulation is used to achieve efficient defuelling of an atomic power station at the end of its working life, within a specified timescale, and ensure a resource smooth project. Working to a tight deadline, a number of operations must be undertaken on a reactor of unique design. The simulation optimises the performance of a discrete series of operations.

This paper discusses the necessity for checking the consistency of the data used in the model, before reporting the model as realistic. Additionally, the validity of the simulation results are checked using cusum techniques.

Through the simulation, it has been shown that the timescale for defuelling this reactor can be reduced from the initial estimate of 24 months to 11 months.

INTRODUCTION
Applications of simulation range from the initial design process to the re-engineering of a fully-functioning manufacturing facility. Simulation is one of the most widely used operations in research and management science techniques, if not the most widely used (Law and Kelton, 2000).

In addition to the traditional use of simulation for high level capacity planning, there are many other benefits including factory layout design, production routing, production mix, throughput prediction, bottleneck identification, new resources deployment. According to Robinson (1994), applications of simulation may be summarised into eight categories: facilities planning, obtaining best use of current facilities, developing methods of control, materials handling, examining the logistics of change, company modelling, operational planning, and training operations staff.

In modern facilities, the available flexibility introduces another degree of flexibility in decision making. The lack of a clear understanding of the dynamics and interaction of components of modern manufacturing systems calls for the use of simulation as an essential support tool.

One of the operational areas which is not widely covered is the case of a declining/closing down company. In many instances, a company may face closure either by virtue of its shortcomings (inadequacy), by the changes made at a corporate level of a multi-site plant or by its natural course. Although closure may not appear a positive process, it may involve significant sums of money (such as in the defuelling and disposing of nuclear power plant reactors). Full understanding of the operations and interactions involved is essential.

This paper discusses the use of a simulation package “Witness” to understand the dynamics in defuelling the Bradwell Power station, the analysis of the defuelling lead times and to assist in optimising the system to achieve a tight deadline. The paper also demonstrates the need for attaining data consistency in the simulation model before conducting performance enhancement.

CASE STUDY

General Background
Bradwell Power Station is a Magnox type nuclear power station, situated on the east coast of England. After 40 years of continuous service, the Station has finally ceased power generation and was preparing to undergo routine defuelling of both nuclear reactors, in readiness for the follow-up of decommissioning activities.

Nuclear Power Stations produce electricity by means of transferring the heat from the uranium fission reaction, to boilers that in turn produce steam to power steam turbines and electrical generators.

The uranium fuel elements are stacked on top of each other, separated by steel and graphite fuel channels, in the reactor
graphite core. A group of 24 Channels form a unique standpipe. Each reactor has 100 standpipes, thus there are 2400 channels per reactor or 4800 channels all together (Figure 2, pile cap area). The uranium metal is cast into rods that are then placed inside a non-oxidising magnesium alloy, called Cage (hence the name Mag-nox). The Cage was designed not only to protect the fuel rods from oxidation, but also to prevent the fission products escape into the coolant gas (CO2). Once the Uranium fuel has been spent, it is withdrawn from the Reactor, and then its cladding (Cage) is removed, compacted and stored on site, as low level waste, in special underground Vaults (Figure 2, Ponds area).

The pressure vessel of each reactor contains 2400 fuel channels. Eventually, these must all be withdrawn from the reactors Once the fuel channels have sufficiently ‘cooled down’ in the cooling Ponds, they will then be transferred into containment flasks and transported to the reprocessing plant at Sellafield.

Overview of Defuelling Operations

The British Nuclear Group has developed a strategic plan for de-fuelling of all its Magnox reactors. The critical feature of this was maintaining a regular flow of spent fuel to the Fuel Reprocessing Plant. There are significant penalties if Bradwell does not complete the defueling within an 18 month time period. Any disruption to this planned flow will have an adverse consequence on other Magnox stations and can result in their earlier close-down. For obvious reasons, the subsequent financial implications for the company would be, in such a case, severe.

The 18 month target for de-fuelling Bradwell was challenging and demanding. To meet these targets, defueling of 48 channels as a minimum between both reactors was required.

The defueling work type comprises of all those activities required to allow safe defuelling of the reactors and the despatch of irradiated spent fuel to the reprocessing plant. The process is summarised in Figure 1 and illustrates the three major plant areas:

Pile Cap area - achieves the withdrawing and the consequent movement of spent fuel from the Reactors to the cooling ponds using a specially designed fuelling/defuelling machine (referred to as the Charge Machine - CMC) that is permanently located on top of the reactor, called the Pile Cap. CMC can be positioned to any standpipe location on the Pile Cap, by crane, rail and gantry. These machines are then coupled, via an intermediary shielding (Chute Head Box – CHB), to the reactor’s pressure vessel. Upon achieving a sealed connection and adequate pressurisation, the CMC will access the reactor’s graphite core, by removing a shielding plug from the standpipe assembly. Once the desired channel within the core is precisely located, the old fuel is removed and put into CMC storage. On completion of this operation, the reactor’s pressure vessel seal is restored, by replacing the shielding plug, and the machine moves off to discharge the spent fuel into a shielded facility (the Discharge Well). Similar sealed connections and disconnections happen at this location as well.

Ponds area - this facilitates the shielding, cooling and preparing of the spent fuel for dispatch. Once the fuel has been lowered into a Basket Skip, already submerged in Ponds water, it is subsequently transferred to a Desplintering Machine. This machine produces the rods to be laid in Fuel Skips, and the Cages to be taken to the compactor and storage.

Flask Handling area - attains the dispatching of spent fuel off-site. At the end of the cooling down period, the Fuel Skips are placed into a transport Flask, which had been also submerged in Ponds water. When ready, the Flask would be dispatched to the reprocessing plant.

SIMULATION MODEL

The simulation package, Witness (Lanner Group, 2004), was used to build a model for the defueling process of Bradwell Power Plant (Figure 2). The initial set up of the process was estimated to have taken approximately 24 months. The aim was to reduce this lead time to 18 months or less.

The model consisted of all the process elements described above, setup, maintenance, breakdown/repair elements of machines and work shifts of all involved operators. The pond maintenance data is shown as an example in Table 1.

Model Data Consistency

In a performance analysis process, it is highly desirable that the data and the process of a given system are predictable. In terms of performance improvement, the system’s variability has to be minimised to an acceptable level before an effective performance improvement process can commence. According to Deming (2000), a process with no indication of special (systematic) causes of variation is said to be in statistical control. It is a random process. Its behaviour in the near future is predictable. A system that is in statistical control has a definable identity and a definable capability. In the state of statistical control, all special causes so far have been removed. The remaining variation is left to chance – i.e., to common causes. The next step is to improve the process. Improvement of the process can be pushed effectively, once statistical control is achieved and maintained, i.e. once the output variability has been minimised to an acceptable level.

Since most simulation models use random variables as input, the simulation output data are themselves random and can only be properly assessed by conducting multiple simulation runs. In most operations systems a lower level of variability is preferred since it is easier to match resources to the levels of demand (Robinson, 2004). Indeed, a worse average with low variability may be selected in preference to a better average
with high variability. Generally, if the variability of the output is high, minimising the variability is recommended prior to considering investment in resources to improve performance.

It is important when simulating a process to ensure that the model is not significantly affected by changing the random number streams, i.e. the variability of the output data is within statistical control. The consistency of the defuelling simulation model output data was checked by conducting 25 runs with different random number streams for each run and for each element and each corresponding data. This was designed to create more realistic randomness to these events. Outputs of the 25 runs were recorded and a Cusum (90% confidence) conducted as shown in Table 2 and Figure 3. The C2 semi-parabolic mask as defined in BS 5703 part 3. It can be seen from the graph, in Figure 3, that the data were all between the upper mask and the lower mask, indicating good consistency. A similar stability check was conducted on the last model and it again showed satisfactory stability.

RESULTS

The simulation run was based on a 24 hour shift; there was no daily start-up or shut down period. Running the base model, as it then existed, recorded that a total defuelling time of 1,028,408 minutes (24 months) was required, which is consistent with the initial estimate. From the simulation report (Figure 4) it became apparent that the Flask was the most hardworking area and it was decided that improvements be made to this centre. The feasible options, as identified by the operational group, were to reduce the setup time by 25 per cent, reduce the breakdown interval by 25 percent, decrease the cycle time by 10 percent and to provide a site storage capability for an additional Flask and its loader. The costs associated with this changes were well within the limits of the budget allocated.

The effect of the above changes on the flask led to a total simulated run time of 534,327 minutes, which is remarkable. As further improvement to the Flask was not achievable, it was decided that improvements focussing on the plant operators (POs) could be a possibility, as their utilisation had significantly increased. Successive improvements on POs finally resulted in an acceptable total defuelling time of about 11 months, as shown in Figure 5.

CONCLUSIONS

A reduction in defuelling time of over 50% has been achieved through efficiencies gained by accurate simulation of the process. By checking the consistency of the input and output data and by further running a cusum techniques test on the outcomes, there can be increased confidence in the validity of the simulation.

ACKNOWLEDGEMENTS

The authors are very grateful to Magnox Electric Ltd. for the data provided for this simulation, for helpful discussions and for permission to publish this paper.

REFERENCES:
Roman S, 2006, Modelling & Simulation for Operations Management – Case study: Bradwell Power Plant, ARU.
Table 1: Ponds Maintenance data (a)

<table>
<thead>
<tr>
<th>Desplittering Machine 01 SET-UP Mode:</th>
<th>Desplittering Machine 02 SET-UP Mode:</th>
<th>Compactor Machine (CMc) &amp; associated equipment SET-UP Mode:</th>
</tr>
</thead>
<tbody>
<tr>
<td>No. of Ops Interval = 24; Time = 10'</td>
<td>No. cf Ops; Interval = 24; Time = 10'</td>
<td>No. of Ops; Interval = 1; Time = 15'</td>
</tr>
<tr>
<td>BREAKDOWN Mode: Operations;</td>
<td>BREAKDOWN Mode: Operations;</td>
<td>BREAKDOWN Mode: Operations;</td>
</tr>
<tr>
<td>Interval: NEGEXP (240,9)</td>
<td>Interval: NEGEXP (240,11)</td>
<td>Interval: NEGEXP (200,13)</td>
</tr>
<tr>
<td>Repair time: RDIST005</td>
<td>Repair time: RDIST006</td>
<td>Repair time: RDIST007</td>
</tr>
<tr>
<td>Labour: 2MF &amp; 1FE.</td>
<td>Labour: 2MF &amp; 1FE.</td>
<td>Labour: 2FE &amp; 2ME &amp; 2EF.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Repair Time in minutes ('Value')</th>
<th>Frequency ('Weight')</th>
<th>Repair Time in minutes ('Value')</th>
<th>Frequency ('Weight')</th>
<th>Repair Time in minutes ('Value')</th>
<th>Frequency ('Weight')</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>30</td>
<td>5</td>
<td>30</td>
<td>5</td>
<td>30</td>
<td>5</td>
</tr>
<tr>
<td>60</td>
<td>31</td>
<td>60</td>
<td>20</td>
<td>35</td>
<td>3</td>
</tr>
<tr>
<td>9</td>
<td>1</td>
<td>90</td>
<td>1</td>
<td>40</td>
<td>8</td>
</tr>
<tr>
<td>240</td>
<td>4</td>
<td>240</td>
<td>2</td>
<td>45</td>
<td>5</td>
</tr>
<tr>
<td>400</td>
<td>1</td>
<td>400</td>
<td>1</td>
<td>60</td>
<td>2</td>
</tr>
<tr>
<td>480</td>
<td>2</td>
<td>480</td>
<td>3</td>
<td>180</td>
<td>3</td>
</tr>
<tr>
<td><strong>1440</strong></td>
<td><strong>1</strong></td>
<td><strong>1440</strong></td>
<td><strong>4</strong></td>
<td><strong>360</strong></td>
<td><strong>1</strong></td>
</tr>
</tbody>
</table>

Figure 1. Defuelling Sequence Chart
Table 2. Consistency Test of the Base Model

| Ctrl Factors | 8.5 | 8 | 7.5 | 7 | 6.5 | 6 | 5.5 | 4.7 | 3.6 | 2.3 | 1 |

![Figure 3. Consistency Test (Cusum chart) of the Base Model](image)

![Figure 2. Witness model of Bradwell Defuelling Processes.](image)
Figure 4. Simulation results of Base Model (Defuelling time = 24 months)

<table>
<thead>
<tr>
<th>Process</th>
<th>% Busy</th>
<th>% Setup</th>
<th>% Broken</th>
<th>% Setup Wait</th>
<th>% Cycle Wait</th>
<th>% Repair Wait</th>
<th>% Idle</th>
<th>% Blocked</th>
</tr>
</thead>
<tbody>
<tr>
<td>Flask24chs</td>
<td>65.28</td>
<td>11.81</td>
<td>1.35</td>
<td>3.61</td>
<td>15.44</td>
<td>0.44</td>
<td>2.07</td>
<td>0</td>
</tr>
<tr>
<td>DischargeWell02</td>
<td>17.23</td>
<td>2.15</td>
<td>0.43</td>
<td>0.09</td>
<td>0.63</td>
<td>0.29</td>
<td>67.86</td>
<td>11.12</td>
</tr>
<tr>
<td>DischargeWell01</td>
<td>17.23</td>
<td>2.15</td>
<td>1.12</td>
<td>0.02</td>
<td>0.48</td>
<td>0.94</td>
<td>69.15</td>
<td>8.92</td>
</tr>
<tr>
<td>Desplittering02</td>
<td>17.23</td>
<td>0.12</td>
<td>0.05</td>
<td>0.13</td>
<td>2.35</td>
<td>0.06</td>
<td>74.28</td>
<td>5.78</td>
</tr>
<tr>
<td>Desplittering01</td>
<td>17.23</td>
<td>0.12</td>
<td>0.11</td>
<td>0.06</td>
<td>2.43</td>
<td>0.02</td>
<td>74.39</td>
<td>5.64</td>
</tr>
<tr>
<td>ChargeMachine02</td>
<td>17.23</td>
<td>2.84</td>
<td>2.62</td>
<td>0.19</td>
<td>0.63</td>
<td>0.68</td>
<td>51.71</td>
<td>24.09</td>
</tr>
<tr>
<td>ChargeMachine01</td>
<td>17.23</td>
<td>2.84</td>
<td>1.51</td>
<td>0.49</td>
<td>1.57</td>
<td>1.36</td>
<td>51.67</td>
<td>23.32</td>
</tr>
<tr>
<td>Compactor24chs</td>
<td>5.74</td>
<td>0.36</td>
<td>0.0</td>
<td>0.35</td>
<td>0.33</td>
<td>0</td>
<td>93.22</td>
<td>0</td>
</tr>
<tr>
<td>FuelSkip02</td>
<td>0.72</td>
<td>0</td>
<td>0.02</td>
<td>0</td>
<td>0.24</td>
<td>0</td>
<td>94.98</td>
<td>4.05</td>
</tr>
<tr>
<td>FuelSkip01</td>
<td>0.72</td>
<td>0</td>
<td>0.01</td>
<td>0</td>
<td>0.06</td>
<td>0</td>
<td>95.38</td>
<td>3.82</td>
</tr>
</tbody>
</table>

Figure 5. Simulation Results of Final Model (Defuelling time = 11 months)

<table>
<thead>
<tr>
<th>Process</th>
<th>% Busy</th>
<th>% Setup</th>
<th>% Broken</th>
<th>% Setup Wait</th>
<th>% Cycle Wait</th>
<th>% Repair Wait</th>
<th>% Idle</th>
<th>% Blocked</th>
</tr>
</thead>
<tbody>
<tr>
<td>DischargeWell01</td>
<td>38.08</td>
<td>4.74</td>
<td>2.48</td>
<td>0.0</td>
<td>0.13</td>
<td>1.22</td>
<td>34.23</td>
<td>19.09</td>
</tr>
<tr>
<td>DischargeWell02</td>
<td>38.08</td>
<td>4.74</td>
<td>0.96</td>
<td>0.02</td>
<td>0.14</td>
<td>1.3</td>
<td>32.6</td>
<td>22.15</td>
</tr>
<tr>
<td>Desplittering01</td>
<td>38.08</td>
<td>0.26</td>
<td>0.24</td>
<td>0.03</td>
<td>0.47</td>
<td>0.25</td>
<td>54.51</td>
<td>6.15</td>
</tr>
<tr>
<td>Desplittering02</td>
<td>38.08</td>
<td>0.26</td>
<td>0.1</td>
<td>0.05</td>
<td>0.53</td>
<td>0.03</td>
<td>54.66</td>
<td>6.26</td>
</tr>
<tr>
<td>ChargeMachine02</td>
<td>33.32</td>
<td>5.24</td>
<td>4.48</td>
<td>0.14</td>
<td>0.63</td>
<td>1.47</td>
<td>3.98</td>
<td>50.73</td>
</tr>
<tr>
<td>ChargeMachine01</td>
<td>33.32</td>
<td>5.24</td>
<td>2.66</td>
<td>0.29</td>
<td>2.18</td>
<td>1.95</td>
<td>3.88</td>
<td>50.47</td>
</tr>
<tr>
<td>Flask24chs</td>
<td>32.63</td>
<td>4.85</td>
<td>0.95</td>
<td>1.6</td>
<td>7.54</td>
<td>0.52</td>
<td>51.89</td>
<td>0</td>
</tr>
<tr>
<td>Compactor24chs</td>
<td>12.69</td>
<td>0.79</td>
<td>0</td>
<td>0.12</td>
<td>0.18</td>
<td>0</td>
<td>86.21</td>
<td>0</td>
</tr>
<tr>
<td>FuelSkip01</td>
<td>0.79</td>
<td>0</td>
<td>0.01</td>
<td>0</td>
<td>0</td>
<td>0.02</td>
<td>99.18</td>
<td>0</td>
</tr>
<tr>
<td>FuelSkip02</td>
<td>0.79</td>
<td>0</td>
<td>0.02</td>
<td>0</td>
<td>0.06</td>
<td>0</td>
<td>99.13</td>
<td>0</td>
</tr>
</tbody>
</table>
HYBRID PETRI NET BASED MODELING AND SIMULATION OF HYBRID PROCESS INDUSTRIES

Mahsa Ghaeli  
Division of Engineering, Science and Computing, Curtin University  
Bentley, WA, 6845, Australia

Parisa A. Bahri  
School of Engineering and Science, Murdoch University  
Murdoch, WA, 6150, Australia

Peter L. Lee  
Chancellorcy, University of South Australia  
160 Currie St. Level 2, SA, Australia

Email: mahsa.ghaeli@postgrad.curtin.edu.au, p.bahri@murdoch.edu.au

KEYWORDS  
Hybrid, batch, continuous, Timed Hybrid Petri net.

ABSTRACT

Many process industries have hybrid production structures, where both batch and continuous processes are involved. Although much research has been conducted on either continuous or batch process industries, there is a distinct absence of research in hybrid process industries. This study addresses the modeling and simulation of such process industries by proposing a Timed Hybrid Petri net as a suitable method. The applicability and potential benefit of the proposed technique is shown through a typical hybrid sugar mill case study. A number of receivers are considered between batch and continuous operations in this sugar mill plant which results in a schedule with no idle time for batch processing units. Using the proposed method, the minimum size for each of these receivers is estimated.

1. INTRODUCTION

While batch processes involve producing a limited quantity of products based on a recipe, a continuous process is an operation performed continuously over time which results in a high volume production of a standard product. In contrast to batch processes where variables may involve integer values, often in the continuous processes, variables describe continuous quantities. Hybrid process industries, which comprise both batch and continuous processes, are widely used. However, little work has been reported in the literature on modeling and simulation of such systems. Most of these works are based on mathematical models.

Petri nets have been also addressed in the literature as an alternate method for the modeling and simulation of hybrid systems whose modeling requires discrete, as well as continuous variables (Champagnat et al. 1998). Different types of Petri nets have been suggested for the modeling of such systems. Among them, Colored Petri nets are most common (Caradec and Prunet 1997; Valentin and Ladet 1993).

A Petri net is a particular kind of bipartite directed graph populated by three types of objects: places, transitions and directed arcs. The dynamics of the model are represented by the movement of tokens, which are located in places and illustrated by either small dots or a real number. Tokens remain in places, travel along arcs, and their flow through the net is regulated by transitions. The firing rules of transitions in a Petri net model make the consideration of the resource use, storage policy and product sequence decisions. Having both graphical and mathematical properties make Petri nets an applicable tool in the process industries. Further information on Petri nets is given in (Murata 1989; Peterson 1981). The application of Petri nets for modeling and simulation of batch process industries is well known (Ghaeli et al. 2005; Gu and Bahri 2002).

In this paper, Timed Hybrid Petri net is introduced as a suitable method for modeling and simulation of hybrid process industries. Hybrid Petri nets are divided into two parts: batch and continuous. Using Timed Hybrid Petri nets, both batch and continuous parts of a hybrid process industry can be represented in the same model.

This paper is organized as follows. First, a brief definition of a Timed Hybrid Petri net is given in Section 2. Subsequently, in Section 2.1, the Timed Hybrid Petri net model of hybrid processes is discussed through an example. Then, the application of a Timed Hybrid Petri net for modeling and simulation of a hybrid sugar mill plant is investigated in Section 3 and the results are discussed in Section 4. Finally, Section 5 is devoted to a brief conclusion.

2. DESCRIPTION OF TIMED HYBRID PETRI NET

Hybrid Petri Net (HPN) is a class of Petri net and is based on the model presented by David and Alla (2001). Hybrid Petri nets have two types of places and transitions-discrete and continuous. In this paper, the places and transitions in HPN are shown graphically as in Figure 1.

![Figure 1: The Representation of Places and Transitions in HPNs](image-url)
For performance analysis, it is necessary and useful to introduce time delays in a HPN. Time can be considered on places, transitions or arcs. By associating time with arcs in a Timed Hybrid Petri net (THPN) model, not only is the time taken in one place considered but also the time duration from one place to another place can be taken into account (Ghaeli et al. 2005). Timed Arc Hybrid Petri net (TAHPN) is formally defined as follows:

Definition 1. Formally, TAHPN is a nine-tuple $\text{TAHPN} = \{P, T, I, O, W, m, h, \text{time}, D\}$, where:

- $P = \{P_1, P_2, ..., P_n\}$ is a finite set of places;
- $T = \{T_1, T_2, ..., T_m\}$ is a finite set of transitions with $P \cup T \neq \phi$ and $P \cap T = \phi$;
- $I : P \times T \rightarrow R^+$ is the input incidence matrix;
- $O : P \times T \rightarrow R^+$ is the output incidence matrix;
- $W : \{P \times T\} \cup \{T \times P\} \rightarrow R^+$ defines the weight of each arc;
- $m : P \rightarrow R^+$ is the set of positive integers whose $i^{th}$ component, $m(p_i), i \in \{1, 2, ..., m_p\}$, shows the weight of the $i^{th}$ place;
- $h : P \cap T \rightarrow \{D, C\}$, this function defines whether a node is discrete (D) or continuous (C);
- $\text{time} \rightarrow R^+$ shows the time on each arc connecting a transition to a place or a place to a transition; and
- $D : P \rightarrow R^+$ is the set of positive integers whose $i^{th}$ component, $D(p_i), i \in \{1, 2, ..., D_p\}$, shows the time of the $i^{th}$ place.

2.1 Timed Hybrid Petri net Model of a Hybrid Process Industry

In a Timed Hybrid Petri net (THPN) model of a hybrid process, continuous transitions represent continuous parts of the process while a batch section is shown by discrete transitions. While the input to a discrete transition can be either a batch or continuous place, the input to a continuous transition can only be a continuous place, unless there is a discrete place which is both the input and output to the continuous transition.

Figure 2 shows a TAHPN model of a simple example with two processes (batch and continuous) and a storage between them. In this example, the product from the batch unit ($t_1$) is processed in the following continuous process ($t_2$). The interpretation of places and transitions in Figure 2 is given in Table 1. The first number in the parenthesis located on arcs represents the amounts and the second number shows the duration time from a transition (place) to a place (transition). This time can be either the operational time or the duration of transferring materials.

<table>
<thead>
<tr>
<th>Places</th>
<th>Interpretation</th>
<th>Transitions</th>
<th>Interpretation</th>
</tr>
</thead>
<tbody>
<tr>
<td>$p_1$</td>
<td>Materials in the batch operational unit</td>
<td>$t_1$</td>
<td>Batch process</td>
</tr>
<tr>
<td>$p_2$</td>
<td>Storage capacity</td>
<td>$t_2$</td>
<td>Transferring from the batch operational unit to the storage</td>
</tr>
<tr>
<td>$p_3$</td>
<td>Storage</td>
<td>$t_3$</td>
<td>Continuous process</td>
</tr>
</tbody>
</table>

3. SUGAR MILL CASE STUDY

A case study based on a typical sugar mill in Queensland, Australia is considered here. The flowsheet of this sugar mill is shown in Figure 3. Boxes with white color represent batch operations, whereas continuous operations are represented with gray color. Triangular shapes represent storage facilities. The input and output of each pan are shown in the flowsheet. The capacity of storages between batch pans and continuous operational units are given in Table 2. All other data for this case study is presented in the Appendix. For 12 hours of operation, there is only one batch pan schedule and it is illustrated in the Gantt chart in Figure 4. As can be seen, there is no idle time between batch pans. This is due to a number of reasons including the consideration of large receivers (Rec1, Rec2, Rec3, Rec9) in this sugar mill plant.
The problem here is to find the minimum size of these storages such that the pans can operate with no idle time for the time horizon of twelve hours. To solve this problem, first, this sugar mill plant is modeled using TAHPN in software developed by Sadrieh et al. (2005). Then, based on plant data, the TAHPN model is simulated and the minimum size of each storage is found by using lp_solve optimization software (Berkelaar et al. 2004).

Table 2: Storage Specifications (Tonnes)

<table>
<thead>
<tr>
<th>Rec1</th>
<th>Rec2</th>
<th>Rec3</th>
<th>Rec9</th>
</tr>
</thead>
<tbody>
<tr>
<td>150</td>
<td>210</td>
<td>150</td>
<td>150</td>
</tr>
</tbody>
</table>

The steps of this algorithm are as follows:

S1: The TAHPN formulation of this plant is carried out with the aid of the developed Timed Arc Petri net (TAPN) software (Sadrieh et al. 2005). The snapshot of the TAHPN formulation of this plant in this software is illustrated in Figure 5.

S2: The schedule is simulated by the integration of the TAHPN firing rules and branch and bound search algorithm.

S3: The schedule is passed to lp_solve and the constraints for each time interval are set up. Since lp_solve is used here to find the minimum sizes of receivers (Rec1, Rec2, Rec3

Figure 3: The Flowsheet of a Typical Sugar Mill Case Study

Figure 4: Gantt Chart of a Typical Sugar Mill Case Study
and Rec9), the constraints in lp_solve are all on these receivers. As a result, for each time interval, the size of each receiver should be greater than the difference between its input and output.

DISCUSSION

The program is run in C, and the resultant minimum sizes of each receiver are shown in Table 3. As can be seen, in order to have the operational units running with no idle time for a horizon time of 12 hours, much smaller receivers can be considered between batch and continuous operational units. There might be possible constraints such as unreliable equipment, which force the designers of this plant to consider large receivers. However, to find the minimum sizes of each receiver using the proposed approach, it is assumed that all operational units work properly.

| Table 3: Proposed Storage Specification (Tonnes) |
|---|---|---|---|
| Rec1 | Rec2 | Rec3 | Rec9 |
| 10.4 | 43.5 | 25.9 | 101.9 |

CONCLUSION

The TAHPN has been introduced as a suitable tool for modeling and simulation of hybrid process industries with both batch and continuous processes. The applicability of the proposed model is demonstrated through a typical hybrid sugar mill case study. The graphical and mathematical properties of Petri nets, make the simulation very straightforward for this case study and by using the proposed method, the minimum size of each receiver is achieved.

ACKNOWLEDGEMENTS

The authors would like to thank the Australian Research Council (ARC), Parker Centre, and Curtin University for their support.

REFERENCES


APPENDIX

The sugar mill plant considered in this paper is divided into two major pars: high-grade operations and low-grade operations. The data for high-grade batch pans is presented in Table A.1, while the data for low-grade batch pans are given in Table A.2. Table A.3 is devoted to the specification of batch fugals where one tonne of massecuite is processed. The capacity of each storage is given in Table A.4. Finally, flowrates of continuous operational units are presented in Table A.5

<table>
<thead>
<tr>
<th>Table A.1: High-grade Pan Inputs (Tonnes)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pan#1 to Pan#4 (3.5 hr)</td>
</tr>
<tr>
<td>Liqueur</td>
</tr>
<tr>
<td>Magma</td>
</tr>
<tr>
<td>A Molasses</td>
</tr>
<tr>
<td>Massecuite</td>
</tr>
<tr>
<td>Output</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Table A.1 Continued</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pan#6 to Rec3 (3.5)</td>
</tr>
<tr>
<td>Liqueur</td>
</tr>
<tr>
<td>Magma</td>
</tr>
<tr>
<td>A Molasses</td>
</tr>
<tr>
<td>Massecuite</td>
</tr>
<tr>
<td>Output</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Table A.2: Low-Grade Pan Inputs (Tonnes)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pan#5 to Pan#7</td>
</tr>
<tr>
<td>B Molasses</td>
</tr>
<tr>
<td>A Molasses</td>
</tr>
<tr>
<td>Massecuite</td>
</tr>
<tr>
<td>Output</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Table A.3: Fugal Specifications (Tonnes/Hour)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input of Massecuite</td>
</tr>
<tr>
<td>---------------------</td>
</tr>
<tr>
<td>Out</td>
</tr>
<tr>
<td>1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Table A.4: Storage Specifications (Tonnes)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rec1</td>
</tr>
<tr>
<td>150</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Table A.5: Flowrates of Continuous Operational Units (Tonnes/Hour)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Continuous fugal</td>
</tr>
<tr>
<td>35</td>
</tr>
</tbody>
</table>
AN ON-LINE PARAMETER ESTIMATION SCHEME FOR A FAULT DIAGNOSIS SYSTEM

C. Angeli
Department of Mathematics and Computer Science
Technological Education Institute of Piraeus
Konstantinopoleos 38, N. Smirni
GR-171 21 Athens, Greece
E-mail: c_angeli@otenet.gr

A. Chatzinikolaou
S. Patsi 62
GR-118 55 Athens, Greece
E-mail: achatzi@otenet.gr

KEYWORDS
Modelling, Fault diagnosis, Hydraulic systems, On line process

ABSTRACT
Models that are used as part of on-line fault detection methods should describe the dynamics of an actual system as accurate as possible using suitably selected mathematical relations and parameter values. This paper presents an on-line parameter estimation scheme that can represent the dynamics of the system and can be used in the case of incomplete state information. This method is used for the estimation of uncertain parameter values of a mathematical model that has been involved in an expert system for online fault diagnosis based on the deviation between the behaviour of model and measurement results.

INTRODUCTION
Modelling information involved in a diagnostic method is considered as an important technique that improves the reliability and efficiency of the fault detection method (Patton et al. 2000, Frank 1996, Sun et al 2003). Models that run in parallel to the dynamical industrial processes should represent the real system as accurate as possible. In consequence, the parameter values of the derived mathematical equations should be chosen in such a way that these equations could describe accurately the behaviour of the system.

Various parameter estimation methods have been presented for the fault detection purpose in dynamical systems as the use of system models linearized about operating points (Reza and Blakenship 1996), least-square methods (Liu 1995), methods using observers (Krisnaswami et al. 1996), expert systems (Isermann and Freyermuth 1991), neural networks (Han and Frank 1998), Kalman filters (Fox and Janocha 2005) and genetic algorithms (Kulworawanichpongp et al. 2005). For the most of these methods the need for highly accurate estimates of the parameters require high computational load that reduces the capability of the method (Chen 1995) when they are applied for on-line estimation of the parameters and as consequence makes them less suitable for on-line systems.

This paper describes a parameter estimation scheme that is particularly suitable for on-line fault diagnosis because of the low computational load requirements and its capability to operate in parallel to the total on-line fault detection method. For the development of the parameter estimation method the DASYLab data acquisition and control software was used. The proposed method is used for the estimation of parameter values that include uncertainty while other parameter values were estimated from analytical considerations. The developed mathematical model was used as a basis for the development of a fault detection system by being incorporated in an on-line expert system that diagnoses real time faults in hydraulic systems.

ON-LINE ESTIMATION OF PARAMETER VALUES

There are two main categories of parameters: First, parameters that are easily defined from the technical characteristics of the hydraulic components or operation related parameters of the system and second, parameters that include uncertainty. The uncertain parameters are mainly related to the mechanical friction, hydraulic leakage or included air in the system.

For the determination of uncertain parameters values $M_r$ (friction torque) and $E$ (oil elasticity) the simulation for a set of values near the estimated was performed, and the simulation results were compared with the online corresponding measurements. The optimal values for $M_r$ and $E$ are the values that minimise the difference between the measurements of the actual system and the model. These values are determined using the integral squared error (ISE) method. The determination of the optimal value for $M_r$ is performed in relation to the commonly used range of values for the parameter $E$ (oil elasticity including air) in order to determine more precisely the optimal value for both parameters. This determination is performed by measuring the integral squared error (ISE) between the measured and calculated signals over a period of time and looking for the minimum value of the ISE according to the following relations:
\[ I_a = \int_{0}^{t_{end}} (p_{am} - p_{am})^2 \cdot dt = F_a(M_r) \rightarrow \min \]
\[ I_b = \int_{0}^{t_{end}} (p_{bm} - p_{bm})^2 \cdot dt = F_b(M_r) \rightarrow \min \]

The integral squared error is measured using the signal analysis capabilities of the DASYLab software by combining two DASYLab experiments. According to this method, the first experiment, Figure 1, performs the control of the actual system and the measurements under various operating conditions and updates the operational parameter values of the input files to the simulation program. In the second experiment, Figure 2, the results from measurements and simulation are compared and processed. Between the two experiments the simulation program runs using the corresponding input data files with the updated parameter values. This method reduces the experimentation time considerably and allows us to perform experiments with a large variety of parameter sets.

The worksheet of Figure 1 consists of three groups of modules. The module group A is responsible for the starting of the actual system. The module group B is responsible for the control of the command voltage \( U_2 \). The module group C is responsible for the data measurement and storing for further processing by the second experiment shown in Figure 2.

Figure 1. Worksheet for experiment control and measurements

In the experiment of Figure 2 the output from model and the output from measurements are compared. After this comparison of measured and calculated data, the integral squared error between them is derived using suitably formulated mathematical and statistical DASYLab modules.
In this experiment the arithmetic module "Pam-Pas" calculates the difference between calculated and measured pressure $p_a$ over the time period $t = 0$ to 0.4 s. The arithmetic module "$(Pam-Pas)^2$" calculates the square of the pressure difference.

The module "Int$(DPa)^2$" calculates the integral of $(Pam-Pas)^2$. The module "Inta0->tend" calculates the value of this integral in the time period $t = 0$ to $t$ (0,4 s). The result is the integral squared error over the observed period of time and is written to the file "integral pa.asc", represented by the module "integral pa". This procedure is performed for a set of values near the expected value of the friction torque $M_f$ and the result is appended to the data of the file "integral pa.asc".

The processing of pressure $p_b$ is performed in a similar way with the corresponding modules "Pbm-Pbs", "$(Pbm-Pbs)^2$", "Int$(DPb)^2$", "Int b 0 ->tend" and "integral pb".

The integral squared difference for the pressures $p_a$ and $p_b$ is the basis for the estimation of the best $M_f$ value. In the DASYLab “experiment” of Figure 3 the files “integral pa” and “integral pb” are processed for the determination of the optimal $M_f$ value.

Figure 3. Determination of the minimum integral squared error for pressures $p_a$ and $p_b$
The module “integ(rm)” displays the integral squared error values for pressure \( p_a \) and \( p_b \) from the files “integral pa” and “integral pb” for various \( M_r \) values. The module “rm/integ” displays these values in a list form. The module “min integ” is a digital meter module that displays the minimum value between the integral squared error values of the list.

In order to estimate an accurate value for the parameter \( E \) this procedure was performed for oil elasticity values of \( 0,90 \cdot 10^0 \), \( 1,00 \cdot 10^1 \) and \( 1,10 \cdot 10^6 \) N/m\(^2\) in the simulation, because these values lie near to the commonly used value for hydraulic mineral oil of \( 10^9 \) N/m\(^2\) if we consider the elasticity of the included air.

The EASYLab experiment of Figure 2 was performed for the pressures \( p_a \) and \( p_b \) with \( p_0 = 50 \) bar, \( U_2 = 6 \) V and oil elasticity values \( E = 0,90 \cdot 10^0 \) N/m\(^2\), \( E = 1,00 \cdot 10^9 \) N/m\(^2\) and \( E = 1,10 \cdot 10^9 \) N/m\(^2\). The results of the minimum integral squared error are plotted in Figure 4.

![Figure 4](image.png)

**Figure 4.** The minimum integral squared error for pressure \( p_a \) and pressure \( p_b \) by \( p_0 = 50 \) bar, \( U_2 = 1 \) to \( 6 \) V and \( E = 0,90 \cdot 10^0 \) N/m\(^2\).

In this Figure it is seen that the value \( M_r = 1,8 \) Nm minimises the difference between measured and calculated pressure \( p_a \) and the value \( M_r = 2,0 \) minimises the difference for the pressure \( p_b \). Therefore the average value of \( 1,9 \) Nm is taken for \( M_r \).

The value \( 1,00 \cdot 10^9 \) N/m\(^2\) for the oil elasticity parameter \( E \) is also the most accurate, because for a slightly lower and a slightly higher \( E \) value the minimum values of the Integral Squared Errors of the pressure differences are higher than for \( E = 1,00 \cdot 10^9 \) N/m\(^2\).

In order to test the performance of the model with the above estimated parameter values and to illustrate the changes of the pressure differences in relation to the operating parameters, the experiment of Figure 2 was used. The maximum differences of pressures \( p_a \) and \( p_b \) between measurement and simulation were calculated from the modules “Pam-Pas” and “Pbm-Pbs” for various command voltage values \( U_2 \) and various \( M_r \) values.

**ACCURACY OF THE DIAGNOSTIC RESULTS OF THE SYSTEM**

The effects of changes in parameter values, used in the model, on the simulation results were examined in order to test the performance of the system. The parameters, as the friction torque \( M_r \), the moment of inertia \( Jm \) and the oil elasticity \( E \) were varied. For a variation of \( \pm 5 \% \), \( \pm 10 \% \) and \( \pm 20 \% \) of these parameters the variation of the simulation results was observed and studied.

The maximum deviations were approximately 0,5 bar for a variation of \( \pm 5 \% \), 1 bar for a variation of \( \pm 10 \% \), and 2 bar for a variation of \( \pm 20 \% \). These variations should not influence the fault detection. Observations indicated a similar effect for changes of the other parameters as well as on the pressure \( p_b \).
CONCLUSION

On-line fault diagnosis requires parameter estimation methods that could respond effectively to time restriction situations. In this paper, a parameter estimation scheme that requires low computational load and is able to operate in parallel to the total on-line fault detection method is presented. For the development of the parameter estimation method the capabilities of the DASYLab data acquisition and control software were used that could be potentially applied for the parameter validation process to many other engineering systems.

According to the suitable parameter values used the model is able to simulate quite precisely the actual behaviour of the system and can responds to the requirements of on-line performance. The experimental results provide evidence of the consistency degree between the behaviour of the model and the behaviour of the system and make the parameter estimation method particularly suitable on-line fault diagnosis systems.

REFERENCES


BIOGRAPHY

C. ANGELI is a Professor at the Technological Education Institute of Piraeus. She holds an MSc in Intelligent Systems from University of Plymouth and a DPhil in Intelligent fault detection techniques from University of Sussex. Her current research interest is real-time expert systems and artificial intelligent techniques in fault detection.

A. CHATZINIKOLAOU was for many years researcher at the Technical University of Athens in the department of Mechanical Engineering and is involved in many large projects of hydraulic systems for drive and control automatic machines. He holds a PhD in Dynamic Behaviour of Hydraulic Systems for Robots. His current research interests are the application of Artificial Intelligent Techniques to Hydraulic Systems.
BLOCK WORLD PLANNING WITH INCOMPLETE INFORMATION
AND SENSING ACTIONS AS LINEAR PROGRAMMING PROBLEM

Adam Galuszka and Krzysztof Skrzypczyk
Silesian University of Technology
Akademicka 16
44-100 Gliwice, Poland
E-mail: Adam.Galuszka@sosl.pl

KEYWORDS
Block world, STRIPS system, incomplete information, linear programming, computational efficiency.

ABSTRACT
In the paper block world environment with STRIPS representation is presented as linear programming problem. It is also assumed that information about state of the problem may be incomplete and its influence to linear programming problem solution is shown. Planning in the presence of incompleteness is much more harder. It causes difficulties in solving real application problems. To increase efficiency of solving such problems a transformation to Linear Programming Problem is proposed. Translation of the STRIPS planning to linear programming is based on assumption of the number of planning steps and mapping conditions and operators in each planning step to variables. Variables are mapped to values between 0 and 1: it corresponds to truth degree of conditions and operators.

INTRODUCTION
In the paper problem environment was modelled as Block World with STRIPS representation. This domain is often used to model planning problems (Nilson 1980, Boutilier and Brafman 2001, Kraus et al. 1998, Smith and Weld 1998, Slaney and Thiebaux 2001, Galuszka and Swierniaik 2003, Galuszka and Swierniaik 2003a) because of complex operator interactions and simple physical interpretation. Starting from 1970s STRIPS formalism introduced by Nilson (1980) is popular for planning problems (Weld 1999). Planning problems are PSPACE-complete in general case (see e.g.: Bylander 1994, Baral et al. 2000), even in Block World environment are not easy (here the problem of optimal planning is NP-complete – (Gupta and Nau 1993)).

Block World environment is chosen for illustration of proposed approach despite it has been considered in the literature for decades (e.g. Nilson 1980). Today this domain can be a representation for logistic problems, where moving blocks correspond to moving different objects like packages, trucks and planes (e.g. Slaney and Thiebaux 2001). The case of Block World problem where the table has a limited capacity corresponds to a container-loading problem (Slavin 1996).

In many real world applications an initial state of the problem given as a set of predicates is, realistically speaking, an overidealization since model and measurement inaccuracies, disturbances and imperfect processing procedures result in the uncertainty in the problem variables. The popular approach to modelling uncertainty in STRIPS planning problems is to treat the initial situation as a set of possible initial states. Such a problem is called problem of planning in the presence of incompleteness (Weld et al. 1998) and is usually much more difficult to solve than ‘complete’ problem: belongs to the next level in complexity hierarchy than corresponding problem with complete information (Baral 2000).

The exemplary situation of block world with incomplete information is shown in figure 1 (some blocks are invisible for sensors so there are more than one possible states).

![Figures 1: Block World with Incomplete Information](image)

In this paper a polynomial transformation of STRIPS problem with incompleteness to linear programming problem (LP) is proposed because LP problems are known to be computational easy (Chacziyan 1979).

STRIPS SYSTEM WITH INCOMPLETENESS

In general, STRIPS language is represented by four lists (C; O; I; G) (Bylander 1994, Nilson 1980):
- a finite set of ground atomic formulas (C), called conditions;
- a finite set of operators (O);
- a finite set of predicates that denotes initial state (I);
- a finite set of predicates that denotes goal state (G).

The initial state describes physical configuration of the blocks. Description should be complete i.e. should deal with every true predicate corresponding to the state. The goal state is a conjunction of predicates. Predicates in I and G are ground and function free. In multi-agent environment each agent defines own goal. This description does not need to be complete. The algorithm results in an ordered set of operators (i.e. action sequence) that transforms an initial
state into a goal situation.

Operators $O$ in STRIPS representation consist of three sublists: a precondition list (pre), an add list (add) and a delete list (del). Formally an operator $o \in O$ takes the form $\text{pre}(o) \rightarrow \text{add}(o), \text{del}(o)$. The precondition list is a set of predicates that must be satisfied in world-state to perform this operator. The delete list is a set of predicates that becomes false after executing the operator and the add list is a set that becomes true. Two last lists show effects of the operator execution in the problem state.

One of the way of dealing with uncertainty in planning is to extend the set of operators with sensory actions (Weld 1994). After executing this sensory action the planner knows whether sensed predicate is true or not in the current problem state. This group of operators has some limitations: they may have no preconditions.

As an example let us consider the problem 1 from planning in Block World environment with 4 blocks (called A, B, C, D). The goal is to decompose the initial state. It is assumed one STRIPS operator that moves the block x from the other block to the table, on(x) means that block x is on another block (or on the table), clear(x) means that there is no other block on block x:

move-to-table(x):
preconditions: on(x), clear(x)
del: on(x))
add: clear(y)

The goal is reached if the following conditions are true:
clear(A), clear(B), clear(C), clear(D)

The sensory action

sensing-block(x):
preconditions: (no precond.)
add: clear(x)

is an action that allows to sense what block is under top block after moving top block to the table (see figure 1).

**STRIPS SYSTEM AS LINEAR PROGRAMMING PROBLEM**

Linear programming problem is defined as:

$$\text{Max } f^*x \quad \text{subject to: } A^*x \leq b$$

$$x \quad Aeq^*x = b eq.$$ 

Following (Bylander 1997) the transformation from planning to Linear Programming is based on mapping of conditions and operators in each plan step to variables. Truth values of conditions are mapped to 0 and 1 for the planning without incompleteness, and to any values between 0 and 1 for planning with incomplete information. The objective function reaches the maximum if the goal situation is true in the last step of planning.

As an example let us consider the problem 1 from previous section. Assume 4 steps of planning (states indexes are: 0, 1, 2, 3, 4). Then the objective function is:

$$\text{Max } (\text{clear(A)(2)} + \text{clear(B)(2)} + \text{clear(C)(2)} + \text{clear(D)(2)})$$

If the goal is reached then the objective function is equal to 4 (4 conditions are true in the goal state).

Constraints for Linear Programming problem are (Galuszka and Swierniak 2004):
- at most 1 operator can be applied in each planning step:

$$\Sigma \text{move-to-table}(x)(i) = 1$$

for even $i = 0, 2, 4,$ and:

$$\Sigma \text{sensing-block}(x)(i) = 1$$

for odd $i = 1, 3$ (after moving to table sensing is performed),
- operator can not be applied unless its preconditions are true:

$$\text{on}(x)(i) \geq \text{move-to-table}(x)(i)$$

and

$$\text{clear}(x)(i) \geq \text{move-to-table}(x)(i)$$

for $i = 0, 2, 4$.

Next group of constraints describe changes of the state after applying an operator. These are equality constraints:

$$\text{clear}(x)(i+1) = \text{clear}(x)(i) + \text{sensing-block}(x)(i)$$

$$\text{on}(x)(i+1) = \text{on}(x)(i) - \text{move-to-table}(x)(i).$$

Additional inequality constraints are needed to express information about real situation in planning problem, i.e. what blocks are found after performing sensory action. Without these constraints the planner will search for the solution taking into account all possible situation resulted from incomplete information $2^n$ possible states for n hidden blocks), what is impossible to present by polynomial transormation to LP problem. These constraints take the form:

$$\text{sensing-block}(y)(i+1) = \text{move-to-table}(x)(i)$$

where block y is the sensed block after moving to table block x.

Finally constraints for variables are needed. The constraints should be mapped between values 0 and 1 what corresponds to truth values of the variables and to binary integer linear programming problem.

Summarizing, the size of LP problem that corresponds to STRIPS problem with sensing action is:
- the number of variables: $3n^2 + 2n$;
- the number of constraints: \(1^* (4*n+1+n-n\text{-}hidden)\), where \(n\) is the number of blocks, 1 is the number of planning steps and \(n\text{-}hidden\) is the number of hidden blocks.

For example 1 \((n = 4, l = 4)\) there are 56 variables of LP problem. Nonzero elements of matrix of equality constraints \((A^* x = b)\) are presented in figure 2, nonzero elements of matrix of inequality constraints \((Aeq^* x = beq)\) are presented in figure 3.

Figures 2: Matrix of Nonzero Elements of Equality Constraints

Figures 3: Matrix of Nonzero Elements of Inequality Constraints

SIMULATIONS

Test cases have been implemented in MATLAB and solved using binary integer algorithm. These problems have been generated randomly for the number of blocks from 10 to 30 and the number of planning steps increased till the goal has been achieved (i.e. all variables that denote clear(x) predicates in last step of planning where equal to 1). The results, the time in logarithmic scale versus the number of blocks, are presented in figure 4.

Figures 4: Efficiency of LP transformation

The great advantage of polynomial transformation to linear programming (LP) problem is a possibility of solving translated problem by polynomial-time complete LP algorithm. This approach reduces computational complexity of searching for the solution, but returns solution only in limited number of cases (this is the cost of reduction). The future work will focus on extending translated LP problem to maximize the group of problems solved properly by polynomial LP algorithm.

CONCLUSION

Translation to Linear Programming allows to efficient search for the solution. That is because planning in the presence of incompleteness is usual at least NP-complete problem, Linear Programming is polynomial-time complete problem and translation from STRIPS to Linear Programming is also polynomial (Bylander 1997). The complexity is not reduced since binary integer algorithm has exponential efficiency, but there is possibility to use classical LP algorithm with polynomial efficiency. The cost of this approach is that algorithm can results in non-interpretable solutions for some initial states (what is followed by assumption N ≠ NP).

This work has been supported by grant No. 3 T11A 026 28 in the year 2006.

REFERENCES

Baral Ch., Kreinovich V., Trejo R. 2000. „Computational complexity of planning and approximate planning in the presence of incompleteness.” Artificial Intell. 122, 241-267
On Conversion and Symbolic Computation of Modelica Declarative Models with Matlab

Dorel Aiordachioae
Emil Ceanga
Gabriel Sirbu
“Dunarea de Jos” University of Galati
Electrical and Electronics Engineering Faculty
Domneasca-47, Galati – 800008, ROMANIA
E-mail: {Dorel.Aiordachioae, Emil.Ceanga, Gabriel.Sirbu}@ugal.ro

KEYWORDS
Physical Modelling, Object-Oriented Models, Simulation, Conversion Tool, Modelica, Matlab, Symbolic Computation.

ABSTRACT

The modelling and simulation activities take a lot of time mainly by conversions of the representations of the models in order to use various simulation environments. For general purposes and equation based models, Matlab is widely used both in academia and industrial world, both theoretical and practical studies. For new technologies, based on physical principles and object technologies, represented by e.g. Modelica, there is a gap between the format of the model and the simulation platforms, which are not adapted to the format of the neutral representations of the physical models. The present work is looking to decrease the gap between the modelling and simulation of such models by converting the Modelica format into Matlab format and - finally - by using a symbolic solver prior to the numerical one. The case study is based on the benchmark of the DC motor, which is simple enough to understand and to follow the considered transformations and quite complex, related to the multidisciplinarity of the necessary phenomena.

INTRODUCTION

In the modelling and simulation environments, a strong relation between the capabilities of solver and the representation formalism of the model is necessary. When the simulation of the model is inadequate or impossible there a transformation tool is necessary to develop and thus to obtain the behaviour of the process by solving the equation based model. More, there are common situations of changing of various models among modellers and users that have different solvers. Thus, a translation of the format must be done and to adapt the constraints of the used formalism to the constraints of the final formalism. An example is referred to the management and conversion of the non-causal to causal models. Obviously, there is a compromise among capabilities of the modelling language and those of the solvers. More or less, the solver is an algorithm which, given a system $S$ of equations, tries to compute a non-trivial approximate solution for $S$. The model and the solver must be considered together, like an assemble and making a matched set. This means the model must be accepted by solver and the solver must be adapted to the model representation. The compatibility should be at the levels of the concepts, which is generically represented in the Fig. 1.

The present work supposes that a model under declarative equations formalism should be solved in order to investigate the behaviour of the considered physical system. Such a case is feasible also when the solver is inadequate, from various reasons, starting with memories limitations or speed of the computations. It is supposed that a Matlab environment is available for use.

![Figure 1: A representation of the model-solver interdependency](image-url)
The declarative input model should respect the syntax of the Matlab language.

The general structure of the modelling and simulation tandem is presented in Fig. 2. The activities considered in the present work are presented on grey back-ground of the figure, i.e. the model transformation and the symbolic processing, all these tasks being developed under Matlab environment.

STATE OF THE ART

Several companies and research institutes are pursuing or planning development of tools based on (Modelica, 2005), (Tiller, 2001). As far we know, the most appropriate similar solutions on our project are MathModelica, (MathModelica, 2005), Simelica, (Simelica, 2004), and those of VISIMOD project, (VISIMOD, 2005).

MathModelica is a powerful engineering tool for virtual model building and simulation of all kinds of physical systems, (MathModelica, 2005). Modelica code conversion is supported, (Fritzson and Bunus, 2002). MathModelica uses two basic parts from (Mathematica, 2005): the computer algebra engine, used mainly for symbolic processing with the advantage of using many features of traditional mathematical notation, and the user interface, which provides the programming interface to the user.

The symbolic computer-algebra capabilities can be applied to the equations of Modelica models. In (Fritzson and Bunus, 2002) an example is presented, by extracting time domain equations from a Modelica model and by transforming the equations in Laplace description. More, MathModelica uses the diagram and visualisation tool Visio of Microsoft and Dymola (A Multi-engineering Modelling and Simulation Environment), from (Dynasim, 2005), as simulation engine.

Claytex has released Simelica, a Simulink to Modelica translator. It enables the automatic translation of Simulink.mdl files into equivalent Modelica models using a new Modelica library, the AdvancedBlocks library, (Dempsey, 2003). It works as both a command line tool so that its use can be incorporated into scripts and also as a windows tool complete with graphical user interface (GUI). The last check of the web site made on February of 2006 does not show anything more about Simelica. It seems the Simelica development was stopped or cancelled.

In (VISIMOD, 2005) the basic idea behind the project is to pool resources for Visualization, MODeling, SIMulation and System Identification at Linkoping University, Sweden. It is developed with specific objective including not only modelling and simulation but also visualization, identification and parameter estimation problems as in (Gerdin, 2004).

The present approach is considering a Modelica input like format to the Matlab modelling and simulation environment, (Matlab, 2005), widely used in universities for education purposes and research, with a restriction of symbolic processing and generation of the solutions. Structural properties and analyses could also be considered.

THE CONVERSION TOOL

Moc2Mat is intended to be a Modelica to Matlab conversion tool. Such a conversion is useful especially for the modelling and/or simulation environment with limited power of computation. The reason of using the Matlab environment is done by the wide spread and using of Matlab as modelling and simulation environment of the model-based equations.

It is supposed that there it is a model in a declarative format: as is the Modelica format. The objective is to convert the neutral and declarative format of Modelica into declarative format of Matlab. The conversion tool is roughly a superposition of the successive actions of simple syntax conversion automata. The implemented functions are presented in Fig. 3.

![Figure 3: Models conversion from Modelica to Matlab format](image)

The tool converts the parameters into variables. It removes all the assigned causalities as part of the algorithm section of the Modelica models. It changes the naming convention of Modelica in something without dots (".").
Table 1: An example of the conversion from Modelica to Matlab format

<table>
<thead>
<tr>
<th>Modelica model (as ‘mof’ file)</th>
<th>Matlab model (as ‘m’ file)</th>
</tr>
</thead>
<tbody>
<tr>
<td>model motor_sim</td>
<td>% motor’s model</td>
</tr>
<tr>
<td>parameter Real sti.u0 = 1;</td>
<td>% parameters</td>
</tr>
<tr>
<td>parameter Real cem.R = 2;</td>
<td>sti_u0 = 1;</td>
</tr>
<tr>
<td>parameter Real cem.L = 0.062;</td>
<td>cem_R = 2;</td>
</tr>
<tr>
<td>parameter Real cem.Kb = 0.0398;</td>
<td>cem_L = 0.062;</td>
</tr>
<tr>
<td>parameter Real cem.Kt = 0.0398;</td>
<td>cem_Kb = 0.0398;</td>
</tr>
<tr>
<td>parameter Real sa.rd = 0.1;</td>
<td>cem_Kt = 0.0398;</td>
</tr>
<tr>
<td>parameter Real sa.l = 0.5;</td>
<td>sa_rd = 0.1;</td>
</tr>
<tr>
<td>parameter Real sa.M = 0.2;</td>
<td>sa_l = 0.5;</td>
</tr>
<tr>
<td>parameter Real sa.Jm = 6.76E-006;</td>
<td>sa_M = 0.2;</td>
</tr>
<tr>
<td>parameter Real sa.b = 5.8E-006;</td>
<td>sa_Jm = 6.76E-006;</td>
</tr>
<tr>
<td>Real sti.pe.u;</td>
<td>sa_b = 5.8E-006;</td>
</tr>
<tr>
<td>Real cem.pe.u;</td>
<td></td>
</tr>
<tr>
<td>Real cem.pm.moment;</td>
<td></td>
</tr>
<tr>
<td>Real cem.pm.omega;</td>
<td></td>
</tr>
<tr>
<td>Real cem.e;</td>
<td></td>
</tr>
<tr>
<td>Real cem.i;</td>
<td></td>
</tr>
<tr>
<td>Real sa.pm.moment;</td>
<td></td>
</tr>
<tr>
<td>Real sa.pm.omega;</td>
<td></td>
</tr>
<tr>
<td>Real sa.J;</td>
<td></td>
</tr>
<tr>
<td>algorithm</td>
<td></td>
</tr>
<tr>
<td>cem.e := cem.Kb*cem.pm.omega;</td>
<td></td>
</tr>
<tr>
<td>equation</td>
<td></td>
</tr>
<tr>
<td>sti.pe.u = sti.u0;</td>
<td></td>
</tr>
<tr>
<td>cem.pe.u = cem.i*cem.R + cem.L * der(cem.i) + cem.e;</td>
<td></td>
</tr>
<tr>
<td>cem.pm.moment = cem.Kt*cem.i;</td>
<td></td>
</tr>
<tr>
<td>sa.J = 0.5 * sa.M * sa.rd * sa.rd;</td>
<td>-sa.pm.moment = (sa.J+sa.Jm)<em>der(sa_pm.omega) + sa.b</em>sa.pm.omega;</td>
</tr>
<tr>
<td>-sa.pm.moment = (sa.J+sa.Jm)<em>der(sa.pm.omega)+sa.b</em>sa.pm.omega;</td>
<td></td>
</tr>
<tr>
<td>sti.pe.u = cem.pe.u;</td>
<td></td>
</tr>
<tr>
<td>cem.pm.moment + sa.pm.moment = 0;</td>
<td></td>
</tr>
<tr>
<td>sa.pm.omega = cem.pm.omega;</td>
<td></td>
</tr>
<tr>
<td>end motor_sim;</td>
<td></td>
</tr>
</tbody>
</table>

It converts all the iterative structures, e.g., if, while, for, from Modelica to Matlab syntax; finally, a file is created with the “.m” extension.

The tool is not perfect yet and requires from time to time some human assistance in the conversion of some particular cases. In future, this drawback will disappear.

The input and the output file of the conversion tool are presented in Table 1, as an example. In order to make simulation, the next step is related to discretization, i.e. the transforming of continuous time variables to discrete time variables. Alternatively, the second step could start with a symbolic pre-processing step, based also on the symbolic toolbox of the Matlab.

**SYMBOLIC SOLVING - AN EXAMPLE**

The Modelica model is took from a “mof” file, e.g., automatically generated by the modelling environment, as Dymola. The file contains the declarative equation model. In modelica, the names of the variables are linked by dots, “.”, showing the inheritance feature of the object-oriented technology of the language. The conversion tool, from Modelica to Matlab, should avoid this convention. In the present work, the dots are replaced by underlines “_”. Let us consider a DC motor’s model with, as internal variables, the current i(t) and the angular speed of the motor’s shaft ω(t):

\[
L \cdot \frac{di(t)}{dt} = -R \cdot i(t) - K_b \cdot \omega(t) + u(t) \quad (1.a)
\]

\[
J_T \cdot \frac{d\omega(t)}{dt} = K_t \cdot i(t) - b \cdot \omega(t) \quad (1.b)
\]

\[
y(t) = \omega(t) \quad (1.c)
\]

The Modelica simplified model is described by the sentences of Table 1. Some details could be obtained, e.g., from [Fritzson, 2004]. The conversion from Modelica to Matlab format is made following the transformations of Fig. 3, which are implemented in Matlab program, designed for conversion. After running, the model from the right side of the table is obtained.

Using the unknown vector of algebraic equations as
\[
XAE = 
\begin{bmatrix}
\text{sti}_\text{pe}_u, \text{cem}_\text{pe}_u, \\
\text{cem}_\text{pm}_\text{moment}, \\
\text{cem}_\text{pm}_\text{omega}, \text{cem}_e, \\
\text{sa}_\text{pm}_\text{moment}, \text{sa}_J
\end{bmatrix}
\]
(2)

the solution of the algebraic equations is

\[
\text{SOL}_{\text{AE}} = 
\begin{bmatrix}
\sin(t) , \text{sti}_\text{pe}_u , \text{cem}_\text{Ki} * \text{cem}_i, \\
\text{sa}_\text{pm}_\text{omega}, \\
\text{cem}_\text{Kb} * \text{cem}_\text{pm}_\text{omega}, \\
\text{cem}_\text{pm}_\text{moment}, 0.5 * \text{sa}_M * \text{sa}_\text{rd}^2
\end{bmatrix}
\]
(3)

if the input variable is

\[
\text{sti}_\text{pe}_u = \sin(t).
\]
(3.a)

The used Matlab code is

\[
\text{SYMS sti}_\text{pe}_u \text{ cem}_\text{pe}_u \text{ cem}_\text{pm}_\text{moment} \\
\text{cem}_\text{pm}_\text{omega} \text{ real} \\
\text{XAE} = [\text{sti}_\text{pe}_u, \text{cem}_\text{pe}_u, \text{cem}_\text{pm}_\text{moment}, \\
\text{cem}_\text{pm}_\text{omega}, \text{cem}_e, \text{sa}_\text{pm}_\text{moment}, \\
\text{sa}_J] ; \\
\text{AE} = \text{eq}(1,1:n\_eq) ; \\
\text{for } i=1:n\_eq, \text{ "n\_eq" means the number of eqs.} \\
\text{sol}_{\text{AE}}(i) = \text{solve} (\text{AE}(i), \text{XAE}(i)) ; \\
\text{end} ; \\
\]

The dynamic variables define the vector

\[
\text{XDE} = [\text{cem}_i, \text{sa}_\text{pm}_\text{omega}] 
\]
(4)

with the symbolic solution

\[
\text{DE}(1) = \text{Dcem}_i = \\
(cem\_pe_u - cem\_i * cem\_R - cem\_e) / cem\_L ,
\]
(5.a)

and

\[
\text{DE}(2) = \text{Dsa}_\text{pm}_\text{omega} = \\
(-sa\_pm\_moment - sa\_b * sa\_pm\_omega) / (sa\_J + sa\_Jm) ;
\]
(5.b)

By renaming the dynamic variables with

\[
\text{cem}_i \rightarrow i \\
\text{sa}_\text{pm}_\text{omega} \rightarrow \omega
\]
(6)

and putting numerical values of the parameters, the symbolic solution becomes

\[
\frac{di(t)}{dt} = 16 \cdot \sin(t) - 32 \cdot i(t) - 0.64 \cdot \omega(t)
\]
(7.a)

\[
\frac{d\omega(t)}{dt} = 40 \cdot i(t) - 0.0058 \cdot \omega(t)
\]
(7.b)

The used Matlab code is

\[
\text{sol}_{\text{AE}} = \text{sol}_{\text{AE}} ; \\
\text{for } i=1:n\_eq, \text{ "n\_eq" means the number of eqs.} \\
\text{sol}_{\text{AE}}(i) = \text{subs} (\text{sol}_{\text{AE}}(i), \text{PAR}(j), \text{PAR\_VAR}(j)) ; \\
\text{end} ; \\
\]

The numeric solution is

\[
i(t) = -0.26 \exp(-t) + \\
0.02 \exp(-3t) + 0.31 \sin(t) + 0.24 \cos(t)
\]
(8.a)

\[
\omega(t) = 12 \exp(-t) - 12 \cos(t) + 9.7 \sin(t)
\]
(8.b)

and graphically represented in Fig.4, over the time range of [0, 5].

![Graph 1](image1)

![Graph 2](image2)

Figure 4: A result from simulation
CONCLUSIONS

The objective of the work was to present a conversion and symbolic computation tool, from declarative format of Modelica modelling language to the declarative format of Matlab. The reason of using Matlab is based on the wide spread in the field of modelling and simulation. The proposed approach uses a simple and feasible tool to convert the Modelica model into Matlab model, and then to make symbolic studies and analysis based on the symbolic tools of the Matlab. There are some limitations concerning the names of the dynamic variables and the format of the differential equations.

FUTURE WORK

To deal with the growing complexity of modelled systems in the any physical modelling language, e.g. Modelica, the next step of the research direction is oriented to the development of debugging tools for simulation models, in the sense of identifying possible causes of errors in the models of the physical systems. Some simple problems might be detected as: insufficient number of equations, non-consistency regarding the measuring units, algebraic loops, identification of the redundant variables, identification and/or elimination of the low dynamic variables, etc. All these operations could be made before sending the model to simulation by a preliminary symbolic manipulation.

Finally, at a more high time horizon, a conversion tool from graphical Modelica format into Simulink graphic format will be considered.

ACKNOWLEDGEMENTS

The authors would like to thank the Romanian National University Research Council for supporting part of the work, under the research grant 1350 / 2004.

REFERENCES

Visio, 2005, Microsoft Inc, USA.

BIOGRAPHY

DOREL AIORDACHIOAIE was born in Dorohoi, Romania and since 1990 is with “Dunarea de Jos” University of Galati, Romania. He graduated the Electronics and Telecommunications Faculty from “Gheorghe Asachi” Technical University of Iasi, Romania, in 1986. His Ph.D. thesis (1997) was developed on neural systems for fault detection and process diagnosis domain focusing on artificial neural signal processing algorithms and methods. The research interests cover metamodelling and modelling and simulation methodologies, in general, with emphasis on object-oriented and neutral modelling technologies.

EMIL CEANGA was born in Reni, Romania and since 1963 is with “Dunarea de Jos” University of Galati, Romania. He obtained his M.Sc. and Ph.D. from Bucharest Polytechnic Institute, Romania, in 1961 and 1969, respectively. He is presently Professor of Electrical Engineering at The University “Dunarea de Jos” of Galati, Romania. His research interests include process modeling, intelligent control techniques and renewable energy systems.

GABRIEL SIRBU was born in Galati, Romania and since 2001 is with “Dunarea de Jos” University of Galati, Romania. He graduated the Faculty from “Dunarea de Jos” University of Galati, Romania, in 1997. He is PhD student on cellular telecommunications area at “Gheorghe Asachi” Technical University of Iasi, Romania. His research interests cover the optimisation methods and algorithms in resources use for cellular communications system.
MODELLING OF TUBULAR AMMONIA DECOMPOSITION REACTORS FOR FUEL CELL APPLICATIONS


*Department of Chemical Engineering and Chemical Technology,
Imperial College of Science, Technology and Medicine,
London SW7 2AZ
Department of Chemical Engineering,
Loughborough University,
Loughborough LE11 3TU
United Kingdom
E-mail: k.hellgardt@imperial.ac.uk

KEYWORDS
ammonia decomposition, model, tubular wall reactor, kinetics, turn over frequency

ABSTRACT
There is an increasing interest in the use of ammonia as hydrogen carrier for fuel cell applications. Ammonia decomposes catalytically into hydrogen and nitrogen without the emission of catalyst poisoning or polluting gases such as CO. Ammonia decomposition in a catalytic wall reactor is modelled and calibrated using actual experimental data in order to compare the derived model parameters such as activation energy, pre-exponential factor and turn over frequency with data from literature. Turn over frequencies in the range from 24molecules.NH$_3$.site$^{-1}$.s$^{-1}$ to 0.31molecules. NH$_3$.site$^{-1}$.s$^{-1}$, depending on temperature, initial concentration and flow rate applied were calculated from the fitted parameters, which compare well with literature values in the range of 4.21s$^{-1}$ to 0.55molecules.NH$_3$.site$^{-1}$.s$^{-1}$ for Ni catalysts. Similarly activation energies of 165kJ.mol$^{-1}$ compare well with data for Ni films (180kJ.mol$^{-1}$) and Ni wires (209kJ.mol$^{-1}$). A number of conclusions are also drawn from the modelling results suggesting that simplifications to the flow and concentrations profiles may be made.

INTRODUCTION
Fuel cells have been in and out of fashion since their invention in the 19th century. Within the last 20 years research on fuel cell technology has expanded exponentially and it appears that enough momentum has been gained by some fuel cell types that they are here to stay (Hirschenhoffer et al., 1996; Kordesch and Simader, 1996). Of the two most promising technologies (polymer electrolyte fuel cells, PEMFC and solid oxide electrolyte fuel cells, SOFC) to emerge, a common denominator is the requirement for hydrogen as a fuel (although developments for external and internal reforming of methanol or methane are also being considered). In this context and also based on concurrent developments in internal combustion engine (ICE) design, the 'hydrogen economy' has been discussed widely. Unfortunately there are severe problems with producing, transporting and storing sufficient hydrogen for this purpose. For the automotive application of fuel cells, hydrogen storage is the ultimate limitation (and challenge). Exotic sorption materials as well as cryogenic storage within the vehicle are considered. Research into a novel hydrogen carrier, namely ammonia is increasingly becoming prominent. When decomposing ammonia according to the following chemical equation (CE 1), hydrogen is released, which can fuel either PEM or SO fuel cells:

\[
NH_3 \leftrightarrow \frac{1}{2} N_2 + \frac{3}{2} H_2, \quad \Delta H = 46.22 \text{kJ.mol}^{-1}
\]  

(CE 1)

Ammonia offers a number of advantages when compared to other fuel alternatives such as pure hydrogen, methanol or methane. Ammonia can easily be stored and transported in the liquid phase at ambient temperature and slightly elevated pressure (8.6bar). Liquid ammonia contains, on a weight bases, more hydrogen than liquefied hydrogen. Recently, Christensen et al., (2004) have demonstrated the storage potential of ammonia even as a solid in the form of a chelated complex with MgCl$_2$. Ammonia leaks are easily detected due to its pungent smell (hydrogen does not smell and would require olfactants to be added). The flammability/explosion limits of ammonia are much higher than those of hydrogen, making ammonia a safer chemical to handle. There would even be a commercial incentive: current ammonia prices are approximately US$ 350 to 375 per ton (Shnitkey, 2003), while the current prices for hydrogen range from US$ 800 to 3,400 per ton (Mann and Ivyy, 2004). There is a scarcity of literature covering the modelling of ammonia decomposition reactors for hydrogen fuel production. Waghode et al., 2005 have investigated the behaviour of a constant temperature wall reactor using finite element analysis. Deshmukh and Vlachos, 2005 have studied two dimensional micro-devices in the form of counter-current combustor/reformers for hydrogen production. Chachuat et al., 2005 take a holistic view at the optimal design of micro power generation units and their modelling, albeit in 1D. This contribution deals with the finite element modelling of tubular, temperature profiled ammonia decomposition reactors (wall reactors) and the subsequent experimental validation of the developed model.

EXPERIMENTAL
Ammonia was decomposed in a pure Nickel tube (99.9%) of 5.1mm 1D and 40cm length situated within a high temperature furnace which exhibits a parabolic temperature profile. Ammonia was diluted in argon in order to be able to model the system using the convection diffusion equation but also to minimise enthalpy effects. The decomposition of ammonia was carried out at
different temperatures (823K to 923K) using different feed mole fractions (5mol-% to 17.6mol-%) and different flowrates (linear velocity from 1.09·10$^{-2}$ to 1.27·10$^{-2}$ m.s$^{-1}$). Effluent concentrations of ammonia, nitrogen and hydrogen were monitored using a Mass Spectrometer (Genesys II, ESS). The ammonia conversion was also determined independently using Flow Response Technology (Richardson et al., 2004). A sketch of the experimental set-up is given in Figure 1.

The temperature profile of the reactor oven, which is required for the detailed modelling of the reactor system, is shown in Figure 2 (for a set temperature of 873K). The temperature profile was fitted using a 2nd order polynomial.

Conversions were calculated on the basis of the disappearance of ammonia as well as the formation of the two decomposition products.

### The Model
Six balances were solved using a finite element package (COMSOL 3.2 – Comsol Ltd.). The underlying balances are:

- the species balance (convection-diffusion equation) for ammonia, nitrogen and hydrogen
  \[
  \mathbf{u} \cdot \nabla C_i - D_{yi} \nabla^2 C_i = 0
  \]  
  \[
  \rho C_p \mathbf{u} \cdot \nabla T + k \nabla^2 T = 0
  \]

Since the reaction only takes place at the reactor walls, this was taken into account by modifying the boundary conditions of the convection-diffusion equation such that specics sources (hydrogen and nitrogen) and species sinks (ammonia) were included. These sources and sinks were coupled by the stoichiometry of the reaction (CE 1) and include the appropriate rate equation. For example, the specics balance for ammonia at the wall of the reactor becomes:

\[
- n \cdot (\mathbf{u} \cdot \nabla P_{NH_3} - D_{NH_3} \frac{d}{dr} \nabla^2 P_{NH_3}) = - r_{NH_3}
\]

In order to simplify the problem and to minimise computation time, use was made of the axial symmetry of the problem, therefore a 2D analysis could be implemented using a cylindrical coordinate system.

**Physico-Chemical Properties**

The binary diffusion coefficients for the problem were calculated using the Fuller, Schettler and Giddings equation (Fuller et al., 1965):

\[
D_{ij} = 0.001 \cdot T^{1.75} \left( \frac{MW_i + MW_j}{MW_i \cdot MW_j} \right)^{0.3} \left( \frac{P}{(DV_i)^{1/3} + (DV_j)^{1/3}} \right)^{10^{-4}}
\]

The temperature dependency of the diffusion coefficients was taken into account in the model, however a mixture diffusion coefficient was not computed because the concentration of reactants and products is rather low.

Therefore the binary data for argon as carrier were used. The following diffusion pairs were equated for a temperature of 873K, Table 1.

### Table 1: Diffusion coefficients for gas pairs at 873K

<table>
<thead>
<tr>
<th>Gas Pair</th>
<th>$D_{12}$ (m$^2$s$^{-1}$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>H$_2$:Ar</td>
<td>4.68·10$^{-4}$</td>
</tr>
<tr>
<td>N$_2$:Ar</td>
<td>1.19·10$^{-4}$</td>
</tr>
<tr>
<td>NH$_3$:Ar</td>
<td>1.48·10$^{-4}$</td>
</tr>
<tr>
<td>Ar:Ar</td>
<td>1.12·10$^{-4}$</td>
</tr>
</tbody>
</table>

As expected, the diffusivity of hydrogen in the carrier gas

![Figure 3: Thermal conductivity of argon as function of temperature](image)
(Argon) is significantly higher than that of the other gases. The thermal conductivities of the individual component gases were obtained from tabulated data in Perry’s handbook (Perry’s, 1984). The mixture thermal conductivity was calculated based on an ideal mixture basis. The thermal conductivities themselves were fitted as functions of temperature (for example, see Figure 3 for the thermal conductivity of argon as a function of temperature) in the temperature range covered by the furnace temperature (300K to 1000K).

The thermal conductivity increases almost linearly with temperature. This is also the case for the other gases. For the purpose of model development a linear regression was fitted to all gas data.

The heat capacities for the single gases at different temperatures were estimated using the Shomate equation (Chase, 1998) and implemented in the model as functions of temperature:

\[ C_P(T) = A + \frac{T}{1000} B + \left( \frac{T}{1000} \right)^2 C + \frac{T}{1000} D + \frac{E}{\left( \frac{T}{1000} \right)^2} \] (7)

\[ C_{P_{mixture}} = \sum_{i=1}^{N} n_i C_{P_i} \] (8)

The gas mixture heat capacity was calculated as an ideal mixture of the constituting components (equation 7). The following Cp values were calculated for a temperature of 873K, Table 2.

**Table 2: Cp values of various gases calculated using the Shomate equation at 873K**

<table>
<thead>
<tr>
<th>Gas</th>
<th>Cp (J/mol.K)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ammonia</td>
<td>53.25</td>
</tr>
<tr>
<td>Nitrogen</td>
<td>31.93</td>
</tr>
<tr>
<td>Hydrogen</td>
<td>29.81</td>
</tr>
<tr>
<td>Argon</td>
<td>20.79</td>
</tr>
</tbody>
</table>

The viscosities of the different gases were also developed as functions of temperature. A linear dependency with respect to temperature was sufficient to describe the viscosities of the single components. The mixture viscosity was again determined on the basis of an ideal mixture. Data were obtained from Perry’s, 1984 and Touloukian, 1975.

**Reaction Kinetics**

The decomposition of ammonia has been studied for a long time and a number of different rate equations have been published over the years. At atmospheric pressure and temperatures in excess of 500K, the equilibrium of equation CE1 lies on the right hand side (Figure 4). The temperature dependency of the equilibrium constant was fitted using an exponential function of the form:

\[ K_{eq} = a \cdot e^{bT} \] (9)

with \( a = 0.4697 \) and \( b = 0.00821 \).

However, most investigators do not take into account the reverse reaction. But, even at low conversions product inhibition by hydrogen must be considered. Bond in ‘Catalysis by Metals’ (1962), discusses an empirical rate equation of the form,

\[ r = k \left( \frac{P_{NH_3}^x}{P_{H_2}^y} \right) \] (10)

\[ k = k_0 e^{-\frac{E_a}{RT}} \] (11)

which is derived from Temkin-Pyshve kinetics.

For a number of transition metal catalysts the parameters in these equations have been determined experimentally. For pure Nickel films Logan and Kemball, (1960) quote: \( x = 0.96, y = -1.53 \). They determined an activation energy of 180kJ.mol\(^{-1}\) and a pre-exponential factor \( k_0 \) of 25.7mol.cm\(^{-3}\).s\(^{-1}\) for a temperature range 663K to 773K. McCabe, 1983 reports activation energies for ammonia decomposition over Ni wires of 209kJ.mol\(^{-1}\). More recently Ganley et al., (2004) have undertaken a comprehensive analysis of the activity of a large number of metallic catalysts for the decomposition of ammonia. For the case of Ni-alumina they quote a TOF of 4.21s\(^{-1}\). Zhang et al., (2005) give a range for \( x = 0.428 \) to 0.318 and \( y = -0.642 \) to -0.477 and a pre-exponential factor of \( k_0 = 1.035 \times 10^9 \) to 3.1610\(^9\)mol.NH\(_3\)/Ni\(^{\mu}\)s\(^{-1}\) and an activation energy of between 95 and 120kJ.mol\(^{-1}\). Choudhary et al., (2001, 2002) quote a turnover number for ammonia over Ni/silica at 873K of 0.55molecules.NH\(_3\)/site\(^{-1}\).s\(^{-1}\). Chellappa et al., (2002) consider a simple model for ammonia decomposition and find an activation energy of 196kJ.mol\(^{-1}\) for Ni-Pt/Al\(_2\)O\(_3\) catalysts in the temperature range of 793K to 933K with ammonia partial pressures of 100 to 780Torr. For an integral reactor (the requirement for a decomposition reactor for fuel cell applications!) the reverse reaction needs to be incorporated into the rate equation and therefore the full Temkin-Pyshve rate equation takes the form:

\[ r = k \left( \frac{P_{NH_3}^2}{P_{H_2}^3} \right)^\beta - \frac{P_{NH_3}^3}{K_{eq}^2} \left( \frac{P_{H_2}^2}{P_{NH_3}^2} \right)^{1-\beta} \] (12)

where the exponents of the partial pressures of ammonia and hydrogen have been replaced by stoichiometric coefficients and the exponent \( \beta \) is the common factor to relate to the empirical exponents \( x \) and \( y \) (equation 7). Zhang et al., (2005) report values for the exponent \( \beta \) from 0.155 to 0.214 for Ni nano-particles supported on alumina. Sattefield, 1991 gives a range of \( \beta = 0.25-0.6 \), albeit for iron catalyst of various compositions.
If we assume that the wall reactor is made of pure nickel and the nickel covers the surface of the reactor wall in a dense packing, then (knowing that the diameter of a nickel atom, 1.38Å) we can calculate the moles of surface atoms available for catalysis per m² to be 8.72×10^5 mol/m². This can now be used to relate to the turn over frequencies (TOF) usually reported in literature and relating these to the surface fluxes incorporated into the boundary conditions of the convection-diffusion equation. These fluxes have the dimension mol.m⁻².s⁻¹.

Initial conditions were set to the boundary data of the inlet (except for the nitrogen and hydrogen concentrations which were set at 0.1 mol/m³). Boundary conditions were: r=0(axial symmetry); Inlet: \( \mathbf{u} = \mathbf{u}_0; T = T_0; C_j = C_{j0} \); Wall: \( \mathbf{u} = 0; T = T_{wall} \); see equation (5) for component wall boundary condition; Outlet: \( P_j = 0 \), \( \mathbf{n} \cdot \mathbf{D}_j \nabla C_j = 0 \).

**Model Predictions**

The finite element model of the reactor setup was run for the different conditions provided for in the experimental section. The model converged after approximately 160 seconds (P4 processor) to yield a stable solution (14 iterations). The mesh had been refined to incorporate 1872 elements. A number of observations can be made at this point. The pressure drop across the reactor is predicted by the model to be 0.271Pa, which is reasonable because the reactor is essentially only an empty tube. The parabolic velocity profile within the reactor is consistent with the laminar flow regime in which the laboratory reactor is working (Figure 5).

![Figure 6: Magnitude of flow field vectors within reactor domain at 873K and ammonia mole fraction of 0.3968.](image)

**Experimental Results**

The ammonia decomposition reactor was run at three different temperatures (823K, 873K and 923K). At each temperature the mole fraction of the ammonia feed was also varied. Conversions were calculated on the basis of the disappearance of the ammonia and also the appearance of products at the exit of the reactor. The ammonia conversions are given in Table 3. It is clear that these are integral conversions (i.e. the reactor was not operated under differential conditions). Therefore the only way to determine the intrinsic kinetics of the reaction over Nickel is the proper simulation of the fluid mechanics and temperature profile within the reactor.

![Figure 7: Ammonia concentration along centre axis of reactor at 873K and ammonia mole fraction of 0.3968.](image)

When considering the entire velocity field within the reactor, it is quite clear that the temperature profile of the reactor has a significant effect on the velocity distribution (Figure 6). The highest velocities are observed in the centre of the tube in the middle of the furnace (that is where the highest temperatures, the set temperature occurs). Due to the small dimensions of the reactor, there is virtually no radial temperature distribution and the temperature profile along the reactor is essentially that of the furnace wall. Although the reaction only takes place at the inner wall of the Nickel tube, virtually no radial
Table 3: Ammonia conversion as function of temperature and feed mole fraction

<table>
<thead>
<tr>
<th>Temperature</th>
<th>823K</th>
<th>873K</th>
<th>923K</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ammonia Mole Fraction</td>
<td>0.177</td>
<td>0.147</td>
<td>0.297</td>
</tr>
<tr>
<td>Ammonia Mole Fraction</td>
<td>0.097</td>
<td>0.19</td>
<td>0.508</td>
</tr>
<tr>
<td>Ammonia Mole Fraction</td>
<td>0.051</td>
<td>0.333</td>
<td>0.61</td>
</tr>
</tbody>
</table>

Evaluation of Rate Constants

The data in Table 3 were used to compare them with the predicted conversions (based on pre-exponential factors, activation energies and orders of reaction (β) extracted from literature). The nine experimental data points above restrict the variable parameters in the model significantly. Table 4 lists the subsequently fitted parameters for the different inlet ammonia concentrations. When a constant order of reaction was applied, activation energy and pre-exponential factor could not be matched to predict the experimental results. It was therefore necessary to consider a changing order of reaction. A change in reaction order has been reported previously (Zhang et al., 2005; Chellappa et al., 2002; Tamaru, 1988). At low ammonia concentrations a first order rate with respect to ammonia tends to be observed (β = 0.5). At high ammonia concentrations it is suggested that ammonia orders much less than one need to be considered.

Table 4: Fitted Parameters

<table>
<thead>
<tr>
<th>C_{ammonia} mol.m^{-3}</th>
<th>E_0 kJ.mol^{-1}</th>
<th>k_0 mol.s^{-1}.s^{-1}</th>
<th>β</th>
</tr>
</thead>
<tbody>
<tr>
<td>7.97</td>
<td>165</td>
<td>4.6×10^6</td>
<td>0.22</td>
</tr>
<tr>
<td>4.37</td>
<td>155</td>
<td>5×10^5</td>
<td>0.3</td>
</tr>
<tr>
<td>2.3</td>
<td>152</td>
<td>1.2×10^5</td>
<td>0.5</td>
</tr>
</tbody>
</table>

The decrease of the pre-exponential factor may be explained by a decreased sticking probability due to the lower concentration of ammonia present. The decrease in activation energy with decreasing ammonia concentration is not very significant and falls within the observed range for bulk Ni (Logan and Kemball, 1960; McCabe, 1983). The change in the value of β points to a change in mechanism. From the pre-exponential factor and the corresponding activation energy, turn over frequencies (TOF) can be calculated. The TOF determined in this work range from 24molecules.NH_3.s^{-1}.s^{-1} to 0.3molecules.NH_3.s^{-1}.s^{-1} depending on temperature, initial concentration and flow rate applied. These results compare favourably with those obtained by Ganley et al., 2004 and Choudhary et al., 2001, 2002.

Conclusions

A finite element method (2D axis-symmetric) was used to model a temperature profiled tubular wall reactor for ammonia decomposition, taking into account the fluid mechanics and the physico-chemical properties of the compounds involved (as a function of temperature). The integral wall reactor requires modelling using Temkin-Psyshov kinetics including equilibrium considerations. Most physico-chemical properties can be fitted by linear or low order polynomial expressions. Due to the high temperatures involved no significant radial concentration gradients are observed. The flow field in the reactor is strongly affected by the temperature profile of the ‘urnace. Calculated turn over frequencies (TOF) compare well with those reported in literature for supported catalysts.

References

Chachuat, B., Mitsos, A., Barton, P.I., Optimal design and steady-state operation of micro power generation employing fuel cells, Chemical Engineering Science, 60, (2005), 4535-4556

Shnitkey, G., Increases in fuel related costs lead to higher production costs, Machinery cost estimates, University of Illinois, USA, (2003)


STEEL MANUFACTURING SIMULATION
The use of simulation to validate a new objective of steelmaking production

A. Bellabdaoui
Service de Mathématique et de Recherche Opérationnelle
Faculté Polytechnique de Mons
9, rue de Houdain 7000 Mons Belgique
e-mail: Adil.Bellabdaoui@fpms.ac.be

S. Allet
Service de Recherche Opérationnelle et de Gestion de la Production
Université de Liège
Bd du Rectorat, 7 B-4000 Liège Belgique
e-mail: Samir.Allet@ulg.ac.be

Abstract
This paper discusses the application of Discrete Event Simulation (DES) for modeling the process of steelmaking continuous casting industry (SCC). The model was developed to validate the increase of steel production to 40%. It was used also to detect possible bottlenecks for the complex system which is characterized by various production rhythms, complex crane movements, etc. A simulation model from blast furnaces to continuous casting continues was developed and analyzed with the simulator Witness. The results of this case study show that we can reach our objective by increasing the steel production to 3.5 millions tons, and detect that the bottleneck of the system is moved from the continuous casting (in the old configuration) towards the converters (in the new configuration).

1 INTRODUCTION AND MOTIVATION
The steel industry is one of the important manufacturing enterprises. The steel is known as the fundamental material in the main industries with the evolution of new manufacturing and construction materials, that’s why the steel industry becomes a dominating factor of an increasingly global market economy [11].

Such a system of production, like all the manufacturing companies, functions with several workshops; each one works with its own rhythm of production and functions as just in time and the provisioning must be synchronized with the principal flow of manufacture. Therefore, the complexity of the flux let the detection of possible bottlenecks a little bit difficult (different rhythms of production, complex movements of cranes,...).

In the setting of the development of the synergies within the Usinor group (become, Arcelor group since 2001) at the time of the purchase of Cockerill Sambre, it has been decided to increase the capacity of production yearly of the hot manufacture of Liege from 2.5 to 3.5 millions tons of liquid steel.

In addition to various adaptations to the production equipments to the blast furnaces and to the steel plant, new investments authorized by Cockerill Sambre in order to increase the production, must be integrated. The principal investment consists in the implantation of one second continuous casting of slabs. This new machine will be reserved to the production of steels with ultra-low percentage of carbon and will be supplied by the existing installation of vacuum treatment in ladle (TSV).

In the new configuration of the steel plant-
continuous casting [9], two paths are put in place, as the figure 1 shows it. The first, including the TSV and the new continuous casting, reserved to the production of steels with ultra-low percentage of carbon (ULC), the second, nourishing the CC1 essentially of low carbon steel calmed with aluminium and some nuances for tin, produces in the existing installation of metallurgy in ladle (MP).

There was an element of uncertainty about how the process would perform. Hence, the manufacturer (plant manager, production manager and engineering manager) decided to undertake a simulation study in order to improve the plant layouts, total production and to identify potential problems. The purpose of the simulation study from blast furnaces to continuous casting continues (CC), is thus two-fold:
- to validate our objective to reach a steel production of 3.5 millions tons; and
- to detect possible bottlenecks in the new configuration.

The simulation was also used to test “what if” scenarios such as increased production requirements, and other effects of operating variables on production.

Since always, simulation has been used as a tool to analyze operations in several areas of steel industry. Models have been used to evaluate production plans and to analyze flux throughout efficiency. Among the recent works applied to the steel-making industry, the papers Bellabdaoui [3] and Bekker [2] discuss a simulation study of crane movement in a steel-making process, the modification of the arrangement of the layout is proposed and is compared to the existing configuration. Wap Sup Um [17] uses the simulator Arena to develop a strategy of control of the different operations of the steel-making process. Thomas and al. [15] presents a variable problem of supply at the boiler which causes failures of a low production in a company of steel wire of high quality. They develop a model of simulation with an objective to increase the flow and to ensure the smoothing of the wire by finding the arrival optimal. Hamoen and Moens [12] develop a steel plant simulator which incorporates a dynamic scheduler to create a realistic production planning. Türkseven and Ertek [16] describe an application of modelling and simulation to estimate and improve quality as well as the performance of the productivity of a company of the steel cord. They focus on the fractures of the wire which can be a significant source of interruption of the system.

The paper is organized as follows. A brief presentation of the art of simulation is reviewed in the next section. Then, a model is ling while taking some consideration to biggest elements intervening in the management of the flux. A simulation result of the model established for the outputs of the converters and the continuous casting is presented next. Finally, the conclusions and a discussion of limitations and future research directions are provided.

2 Simulation methodology

The concept of computer simulation has existed for almost 40 years and has been one of the most powerful tools in the design and analysis of new and existing complex manufacturing systems [13]. Also, simulation is one of the most widely used analytical techniques used by professionals in Operations Research and Management Science [10]. Simulation is more effective than analytical solutions for complex models, where the state of the system changes over time, and the experiment on the actual system may not be possible.

Three typical application areas [1] can be identified:
- explorative studies of existing systems to improve them;
- studies of existing systems with some changes made to them, similar to the first purpose but
used to validate a specific alternative; and
- design and validation of new systems.
In practice, simulation projects are often a combination of these three applications.

Simulation modelling involves the process of abstraction, construction of the model and conducting experiments. Simulation is usually performed on a computer using special software. Moreover, the computer animation generates a moving picture of model operations and can provide valuable insights into the model behavior.

The steps to be followed in simulation modelling involve:
- problem definition;
- conceptual model formulation;
- determination of data requirements and collection of data;
- model translation;
- verification;
- validation;
- experimental design and experimentation;
- analysis of results and interpretation.

The reader is referred to [1], [14], and [10], for literature on simulation and more details of the above steps, particularly discrete event simulation applied to manufacturing systems.

3 System description

The global system is located in the site of Liege in Belgium. The system is dispersed on a way of approximately 24 km, where the blast furnaces are located at Seraing and Ougree, on the other hand the steel-works is located in Chertal (Lieve). In this study, there are three competing workshops, disposed in series, to be simulated as follows:

(1) The workshop 1 presents flux matter of the pig iron that is transported from the two blast furnaces to the entry steelworks in special trucks “wg” then poured into a ladle “pf”. Iron is extracted from its ore in blast furnaces HF6 and HF8. The pig iron is transported to the steelworks in special trucks, which circulate on the National Railway Company of Belgium SNCB.

(2) The workshop 2 assures the transfer of iron ladles “pf” between the stands of tipping and the converters. The second system is made of:
- (1) two identical stands of tipping, (2) two identical stands of desulfurization, (3) three stands of converters, and (4) a movement crane, which moves the iron ladle inside the workshop.

(3) The workshop 3 assures the transfer of steel ladles “pa” between the converters until the continuous casting. It is composed by three stages:
- first, Crude steel is obtained in the converters and poured into a ladle “pa”. Second, the full steel ladles towards the secondary metallurgy (MP or TSV) to adjust the chemical composition of steel following desired quality. Then towards continuous castings (CC1 or CC4) to solidify under the form of slabs.

Important structural assumptions are noted below:
- Engines, travelling cranes and carriages ensure transfers of entities wg, pf and pa on the various workshops;
- two over three converters are active, each converter is regularly maintained; Others machines TSV, CC1 and CC4 undergo times of set-up;
- on the machines of continuous casting, one treats the charges of the same sequence uninterrupted (without: idle period), from where the name of continuous casting;
- between two successive sequences on a machine CC, a time of set-up is envisaged;
- the processing time of a charge is between a minimal value fixed by the possibilities of machine CC and a maximum value fixed by the quality of steel of the charge. Obviously, to ensure the
continuity of casting, it will be necessary to slow down the CC;
- for reasons of quality, time between the end of treatment on the machines converter and beginning on machines CC, does not have to exceed a well defined maximum duration.

The data assumptions that should be noted here are the following:
- in the workshop, 41 wg, 3 pf and 7 pa are used and they are dispersed on the various elements;
- twenty ways of SNCB are programmed each day of which the half on the each steelworks - blast furnace section (files TrajKC and TrajSC);
- the order books consists of two ordered of sequences which are initially introduced into flow metal of the steel-works to obtain:
  - the assignment of the charges starting from the converters as well as nominal times of casting (with deceleration possible);
  - the rhythm of work of the converters (i.e. delays in order to satisfy a constraint of sojourn time).
And thereafter, we are recovered the three files Ecart1, Ecart2 and Flux.

4 Elements of conceptual model

In this study, a conceptual model of the existing SCC process was constructed according to the physical and operational data collected from the system. The SCC process is categorized as a “discrete”, “dynamic”, and “stochastic” system with modular aspect.

The first and more important step is to define the elements that a simulation model needs to work with (see table 1).

Entity:
There are three types of entities in the system: trucks (wg), iron ladles (pf) and steel ladles (pa). In addition to these three significant entities, the model uses a virtual entity ( fictif) which makes it possible to overcome some problems of modelling.

Attributes:
An attribute is an element representing a characteristic of an entity, a machine, etc. For example, the entities wg, pf and pa use a whole of attributes to specify the weight (poids), the capacity (cap) and the number (num) of the entity. Other attributes are related to the entity pa; it is about number of the sequence (NumSeq), number of ladles in the sequence (NumPoche), its destination towards the secondary metallurgy (DestMS) and continuous casting (DestCC), the deceleration of ladles envisaged in the continuous casting (rease).

Machines:
We will note most significant:
- simple machine as HF6, HF8, Desulphurization, MP, TSV.
- machine with multiple cycle as tipping, converter, CC1 and CC4.

Stocks:
Stocks in our model is very varied and in great quantity. They are composed of three types:
- real stocks; it is the case the stations dedicated of each blast furnaces, the storage section at the entry of the steel-works of the empty and full trucks;
- stocks dedicated to installations; as the positions which are with the top of the machines (Desulphurization machine for example);
- stocks which represent a virtual element or a position of waiting to release an entity which is treated as an element or on standby of a mean of handling.

Vehicles and tracks:
Great number of vehicles are modelled which make it possible to move the various articles. The engines of the blast furnaces, train SNCB, the engines at the entry of steel-works, crane travellings and the various carriages model all by vehicles. The tracks are certainly more numerous than the vehicles since they are distinguished in tracks from loading, of unloading and tracks of transfer to go to and others for the return.

Variables of decision:
Model of simulation is composed of several types of variables integer, real, of the names and the character strings. A whole of these variables is necessary for the description of the constraints of some elements. For the machines, the variable
Table 1: Elements of conceptual model

<table>
<thead>
<tr>
<th>Entity</th>
<th>Attribute</th>
</tr>
</thead>
<tbody>
<tr>
<td>wg (47)</td>
<td>Cnf, Pd, Num, Num, Seq, Num, Poche, DestMS, DestCC</td>
</tr>
<tr>
<td>pf (15)</td>
<td>jCpl</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Machines</th>
</tr>
</thead>
<tbody>
<tr>
<td>HP, HPB</td>
</tr>
<tr>
<td>Densification (2)</td>
</tr>
<tr>
<td>Converters (3)</td>
</tr>
<tr>
<td>MP, TSV</td>
</tr>
<tr>
<td>CC1, CC4</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Stock</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vente, Etale</td>
</tr>
<tr>
<td>Vente, Rente</td>
</tr>
<tr>
<td>Garo, Congres</td>
</tr>
<tr>
<td>...</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Vehicle</th>
</tr>
</thead>
<tbody>
<tr>
<td>Loco, HP</td>
</tr>
<tr>
<td>Loco, Selle</td>
</tr>
<tr>
<td>TrainKC</td>
</tr>
<tr>
<td>TrainCC</td>
</tr>
<tr>
<td>E1al</td>
</tr>
<tr>
<td>MS, CCs</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Variables</th>
</tr>
</thead>
<tbody>
<tr>
<td>NbPochesCV</td>
</tr>
<tr>
<td>NbPochesCC</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Resource</th>
</tr>
</thead>
<tbody>
<tr>
<td>Oper, HP</td>
</tr>
<tr>
<td>Oper, HPB</td>
</tr>
<tr>
<td>Oper, Entree (2)</td>
</tr>
<tr>
<td>Oper, Pochers</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Files</th>
</tr>
</thead>
<tbody>
<tr>
<td>TrajKC</td>
</tr>
<tr>
<td>TrajCC</td>
</tr>
<tr>
<td>Ecart1, Ecart2</td>
</tr>
<tr>
<td>Flux</td>
</tr>
<tr>
<td>Stat, CV, ...</td>
</tr>
</tbody>
</table>

5 Data collection and analysis

All data were collected from the existing system to build the conceptual model with new investments. Modifications were to be made to construct the models of alternative systems. All the operations of machines, personnels and transfers times of the vehicles were measured and used to build all the models or the simulation study of this article.

The process data were collected from production reports, and personal observations.

6 Coding

A number of languages can be used for writing computer codes for simulation purposes and building a series of models. We used the visual interactive discrete event simulation system Witness in the present study by consultation the reference of manual [18] and relevant papers written by Calinescu and al. [8] Bellabdaoui [4]. Witness was chosen for this research study because it is an industry standard discrete event system with a large academic and industrial companies. It is characterized by its ability to represent the complex system logic accurately and for its capacity to handle the very large volume of entities moving through the system.

The software is unique in that it enables users to develop models of increasing complexity from a simple three-phase modelling premise. First, models are developed in Witness by defining all of the model elements, e.g. buffers, machines, conveyors, etc., then by adding more detail to the model and, finally, by displaying the salient aspects of the model on the computer screen.

7 Model verification and validation

The simulation model is built stage by stage and is carefully examined in each stage before continuing to the following one. The combination of the stages as a whole was checked at the end. All the entities are correctly forwarded to the elements and all the process have been appropriately defined in order to keep track of the similarity of the simulation model to the existing workshop. The throughput in the simulation model is carefully examined,
and some calibrations were conducted by making adjustments of input parameters by using the observed data and the malfunctions report. This iterative process was continued until the model was considered to be sufficiently accurate. After the validation of the model, the other four models were developed from the modification of this model.

8 Experimentation with the models

The simulation study consists of two types of experiments, each for a competing model. The objective of each experiment was to study the impact of some parameters on the system behavior as a sensitivity analysis. The experiments are related, on one hand to the outputs of the converters and on the other hand, to the outputs on continuous casting.

8.1 Outputs of the converters

Assumptions:

We carried out a simulation on the combined part of workshop 1 and 2 per one 10 days period by taking account of the following assumptions:
- maintenance implemented on the 2 converters;
- the blast furnaces produce sufficient the pig iron: approximately 6000 T each one;
- times of transfers of the two engines carrying out displacements at the entry of the steel-works are variable in the interval of [30 sec, 1min].

Simulation and results:

We simulated two situations:
- \( R1 \): pessimistic situation, times of transfers with their greater value (1 min);
- \( R2 \): optimistic situation, times of transfers with their smaller value (30 sec).

The result of simulation thus obtained by Witness are given in table 2. We note that:
- in the best of the cases, the converters produce approximately 61 ladles/day as an average;
- a simple variation of a parameter of the problem (times of transfers of the engines which go from 30 sec to 1 min) can make the production fall down to 58 ladles/day, as an average;
- overall, the level of production of the converters will have to make it possible to ensure the objective to reach the production of 3.5 tons million of steel.

The results obtained are in conformity with the forecasts of production awaited by the plant manager for Cockerill Sambre and who are of 60 ladles in converters.

<table>
<thead>
<tr>
<th>Sc.</th>
<th>Available</th>
<th>Occupied</th>
<th>Maintenance</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>CV1 : 4.43%</td>
<td>CV1 : 89.32%</td>
<td>CV1 : 6.25%</td>
</tr>
<tr>
<td></td>
<td>CV2 : 4.61%</td>
<td>CV2 : 89.14%</td>
<td>CV2 : 6.25%</td>
</tr>
</tbody>
</table>

8.2 Outputs of continuous casting

In this part, times of transfers of the engines considered will be 30 sec (optimistic situation \( R2 \)). Simulation relates to an order book of 2 sequences on CC1 and 2 sequences on CC4 which are repeated over one 10 days duration.

The assumptions taken during simulation are:
- the blast furnaces produce sufficient the pig iron: approximately 6000 T each one;
- the tests carried out do not regard maintenance converter.

The objective is thus of knowing the limits of production of continuous casting, by carrying out three tests:
- \( R3 \): situation without maintenance converter - rhythm of flow metal: limited to the information provided by the program of flow metal.
- \( R4 \): situation without maintenance converter - modified rhythm of flow metal: in addition to the information provided by flow metal, we introduced on the level of the converters a few processing times higher than those indicated into flow metal.
- \( R5 \): situation with maintenance converter - modified rhythm of flow metal: as the converters maintenance are not programmed in
flow metal, we tried to introduce them in a pragmatic way. Thus, it was possible to insert, at the beginning of each sequence, we added a sufficient time (30 min) to allow a possible maintenance of the converter.

**Remark**: For the news scenarios, it seems reasonable to integrate the same data provided by the program of flow metal for evaluating the production impact.

**Simulation of the model and results**: Table 3 summarizes the results obtained and they are in conformity with waiting of the plant manager of Chertal (Arcelor - Cockerill Sambre). We also obtained, for each situation, the following specific results:

- **R3 situation**: no rupture of continuity on two continuous casting was observed;
- Sojourn times of some ladles, in particular those intended for CC4, exceeding 120 min.

- **R4 situation**: no rupture of continuity on two continuous casting was observed except some rare times appear on the ladles intended for CC4 at the beginning of simulation due to initialization;
- all the sojourn times of the ladles are lower than 120 min.

- **R5 situation**: no rupture of continuity on two continuous casting was observed except rare times appear in end of the ladles intended for CC1, from where the possibility of slowing down in a practical situation;
- all the residence times of the ladles are lower than 120 min.

The occupancy rates show that the bottleneck of the system is at the level of the converters (the sum of the maintenance and occupancy rates exceeds 95%). Other elements (cranes, machines) do not seem to be critical points.

### 9 CONCLUSIONS

This paper concentrated mainly on predicting process performance with complex systems, such as steel manufacturing, which this problem can best be solved with simulation. The purpose of the study was to develop a detailed simulation model of the global system from blast furnaces to continuous casting to validate the increase of steel production to 40% and to detect possible bottlenecks in the new configuration. Experimenting with simulation can mean testing a number of scenarios. Examination of the results reveals that:
- our aim which is to reach a production of 3.5 tons million of steel is validated;
- the bottleneck of the system is moved of continuous casting CC1 (in the old configuration) towards the converters (in the new configuration).

In addition, the model of simulation depends on the use of planning for the workshop 3 (a program owner with Cockerill Sambre). However, we worked out two alternatives of decision-making aid for the planning of the workshop 3. The first is a heuristic method based on elimination of the conflicts between machines [5] and the second is a mixed-integer linear programming model [7]. Henceforth, the objective will be to make a coupling between the model of simulation which is able to take into account the finest constraints of the workshop with a scheduling model (who allows to replace the program flow metal) which is capable to find a planning of the system while taking in consideration the very big combinative of the problem.

**Acknowledgment**
This work of simulation is within the framework of an industrial project named “3.5 millions of Chertal” in collaboration between Mathematics and Operational Research department of the Mons Engineering Faculty and Methods, Information
Systems and Quality Assurance division of the company Arcelor Cockerill Sambre. The project of simulation was developed in collaboration with my colleague S. Allet and it required one year of work since December 1999 to January 2000.

References


Coupled heuristic and simulation for steel production planning

A. Bellabdaoui
Laboratory of Mathematics and Operational Research
Faculté Polytechnique de Mons
9 - rue de Houdain, 7000 Mons, Belgium
e-mail: Adil.Bellabdaoui@fpms.ac.be

Abstract

This paper illustrates the integration of a discrete event simulation and a method of optimization for a steel-making industry. The schedule is capable to find a predictive planning of the system while taking into consideration the very big combinative of the problem, this module is solved with the help of calculation scientific software Matlab. The simulation model is capable to take into consideration the finest constraints of the system (different crane movements and vehicles, etc.), and it was used by the software Witness. The approach of coupling, that is made with Visual Basic, will be used to validate the capacity production as well as to generate a feasible schedule of the process.

1 INTRODUCTION

The integration of scheduling and simulation has become nearly ubiquitous in practice, so that discrete event simulation packages now include some types of optimization routine.

On one hand, it is clear that the methods of optimization are characterized by taking into consideration the very big combinative of the problem, and thereafter they are limited not only by their requirements in computing times and memory capacity, but also by the difficulty in representing the real operation of a workshop of production.

In other hand, the simulation model is a powerful tool to describe the logic of operation of any production processes and it is capable to take into consideration the finest constraints of the complex system (different crane movements and vehicles, ...). In order to take into account these aspects, the adopted solution consists of coupling the simulator with a schedule routine.

The use of methods of optimization coupled to a model of simulation has been applied in different continuous process industries such as steel [1], coffee [12], chemical [2], paper [10] and others manufacturing companies ([8], [9]). However, it was shown that the co-operation (hybridization) of methods (optimization, simulation, ...) makes it possible to integrate a great number of tools complementary for common goal providing a decision-making aid also complete, precise and robust as much as possible.

The context of this study is the installing of a new machine of continuous casting for Arcelor Group in order to increase the annual output of the hot factories of Liege from 2.5 to 3.5 million tons of molten steel. One of the principal logistic objectives is to model and simulate liquid flows from the blast furnaces to continuous castings in order to validate the desired production and to detect possible bottlenecks. However, a part of the simulated model [4] depends on the use of planning of the workshop from the converters with continuous castings (a program specific to the company). However, we worked out two alternatives of decision-making aid for the planning of the workshop. The first is a heuristic method based on elimination of the conflicts between machines [3], and the second is a mixed-integer linear programming model [6].

And thereafter, the objective of this paper is to present an integrate model combining the simulation model with the scheduler, based on the heuristic method which makes it possible to replace
the programme of flow specific to the company.

This section introduced the topic and defined the scope of this paper. The next section describes the industrial considered problem. Section 3 illustrates our approach for combining schedule and simulator. The final section draws conclusions.

2 Industrial considered problem

The industrial problem in which we are interested here, concerns a workshop of an iron and steel industry: continuous casting steel-works. The steel-works produce a large variety of slabs, characterized by the size (width and thickness) and the chemical composition.

2.1 Structure and operation of the steel-works

This workshop is composed of three stages of treatment of flow matter (Figure 1):

1. Two identical converters (CV).
2. Two different stands of refining (RS).
3. Two different continuous castings (CC).

These stands of refining give to the charges the desired metallurgical composition. The two stands of refining respectively treat charges intended for two continuous castings (third stage); again means of handling move the ladles towards the entry - called "swivel" - continuous castings with the help of a constant transfer time. The processing time of a charge is identical, but differs according to the stand from refining.

At the entry of continuous castings, molten steel must be maintained at constant temperature. This implies a sojourn time limited of a charge, between the end of the treatment of the converters and the beginning of the treatment with continuous castings.

Molten steel will be solidified in the form of slabs in contact with walls water-cooled. The charges of the same sequence (cf 2.2 below) must be treated in a continuous way; this continuous flow is ensured by the swivel which contains two charges, one in a state of flow, the other lends to succeed to him by swivelling of the swivel.

The processing time of a charge to continuous casting depends on the charge; moreover this time is a variable taking its values in a certain interval. Indeed, in order to ensure the continuity of the flow, the speed of this one can be modified within certain limits. Another way of ensuring the continuity of the treatments of the charges of the same sequence of this stage, is to delay the beginning of the treatment of the first charge of the sequence. However, this delay is limited by a maximum value, compared to the date of availability of continuous casting.

When two sequences are treated consecutively on same continuous castings, there is an intermediate time of preparation called "set-up" time. This time is variable in a certain interval.

2.2 Sequence and constraints to be respected

A sequence is made of a certain number of charges, to treat in a given order. A sequence is dedicated with one of the two lines of production (RS - CC). In short, see the point 2.1, the hard constraints to be respected of the charges of the same sequence are bellow:
any converter can be used for the first stage; the processing time is constant there;
- there is constant transfer times between stages 1 and 2, and stages 2 and 3;
- the sojourn time of a charge between the end of the treatment at stage 1 and the beginning of the treatment at stage 3 is limited. This constraint is due to the necessity of keeping a sufficient temperature before the charge is processed in the CC;
- the beginning of the treatment of the first charge of the sequence at stage 3 must take place in a window of time;
- there cannot be idle period between the treatment of two consecutive charges of the sequence at stage 3 (continuous casting);
- the processing times of the charges at stage 3 are variable, taking their value in an interval of time. The minimal value is known according to the function of the weight, the width and the maximal speed of the flow of the charge into the CC. The maximal value depends on the quality of the product.

### 3 Combining schedule and simulator

#### 3.1 A general schemas

The overall structure of 3S (Steel Scheduling Simulation) is shown in Figure 2. The 3S consists of three modules: user-interface, scheduler and simulator model.

The user interface is the main program which is implemented with Visual Basic. It plays the role of Master and it will make possible a dialogue between the two modules of schedule and simulator through OLE-automation.

1. Data transmission of the sequences of the main program towards the program of scheduling.
2. It presents the planning of the steel-works from the converters to continuous castings with Matlab Software. It plays the role of the prediction.
3. Transmission of the planning prediction from the main program towards the simulation program.
4. It presents the simulation model with Witness of the steel-works. Its role is a corrector of flow metal by considering all the constraints of the steel-works. It transmits planning carried out to the main program.

#### 3.2 Schedule Routine

The heuristic algorithm [3] consists of three phases. The first one is a simple initialization of the two sequences at stages 2 and 3, taking into account the availability dates of the two CC and the transfer time. The sequences are scheduled at the earliest dates using the minimal processing times of the charges. Such schedule may eventually generate some overlappings at stage 2. The second phase will also treat each sequence separately at stages 2 and 3. The schedule will be modified in order to eliminate the possible conflicts at stage 2 and to respect the maximal sojourn constraint. During this phase, some advances will be first introduced for some charges at stage 2. If these advances are too large with respect to the maximal sojourn time constraint, it will be necessary to slow down some charges, i.e. to increase their processing times at stage 3. The third phase concerns the assignment of all the charges of both sequences to the two CV at stage 1. The charges are considered one by one, in chronological order from their starting time at stage 2. If the availability of the CV does not permit to assign a charge:

- either, if possible, all the sequences of this charge will be delayed,
- or, a new slowdown will be introduced for some charges belonging to this sequence.

In both cases, because of the modification of the schedule at stages 2 and 3, phase 3 must be repeated from the beginning. The heuristic stops when all the charges can be assigned to the CV.

The heuristic algorithm is implemented in Matlab and it begins by download the two sequences, each one dedicated to a particular (RS-CC) pair,
and the initial state of the system, (i.e. the availability of machines CV and CC, sequence which is currently being processed at the initial instant).

The objective of the manager is to treat the two sequences as rapidly as possible, that is, to minimize the completion time of both sequences.

The experimental results show that our heuristic gives interesting results compared with the solution provided by the company.

### 3.3 Simulation model

The simulation model has been built with Witness. The model is presented in [4], [5], and it is categorized as a “discrete”, “dynamic”, and “deterministic” simulation with modular aspect. The very important factor in our simulator is characterized by its higher level of detailed representation, the control logic between different elements, and the finest constraints of the system (set-up of machines, cranes and vehicles, personnel operation, ...), taking the role of the real production system.

### 3.4 Coupled Approach

The graphic interface of our platform, controlling the two modules schedule and simulator, is described in the Figure 3.

The user of the control platform initializes two sequences (each one dedicated to one CC) which are selected from a database, and he gives one state of the system (Figure 4.a). The result of schedule is planning of the system which determines the operation products on machine tools (Figure 4.b). Specific database result, necessary for the execution of the simulator module, can be given in this level; and it represents the routing operation and the decelerate processing time of charges to the converters and the continuous casting. The discrete event simulator serves as a benchmarking system to evaluate the schedules on a richer model (Figure 4.c). This main program was designed to extract data from the result database of simulation, and it will carry out a phase of validation for all the charges in chronological order. This one will check continuity on the level of the CC. The information relating to the not validated part is retransmitted towards the schedule module.

This approach has a great importance for the assistance of a user. One of the originalities of our...
work resides to validate the capacity planning as well as to generate a feasible and real schedule for the remaining sequences with a new state. It enables him also to insert the new or rush sequences.

The model was solved by using a PC Pentium III, 551 MHz with 768 MB of RAM, and the operating platform Windows XP. The computing time is expressed in seconds and the time execution of the model is related to the structure of each example. In general, it is solved on the simulator and the optimizer less than 1 min and 10 sec respectively.

4 CONCLUSION

In this article, we presented an approach of integration of scheduling and simulation of a concrete industrial problem. The basic idea of the approach is to connect the scheduler into a discrete event simulator in order to validate a complete planning and to generate a feasible schedule for the remaining sequences.

This integration is able to provide invaluable tools, adapted to the industrial problems in spite of their limitations in term of theoretical results (proof of optimality ...). Moreover, one such “deterministic” approach can seem restrictive with the first reflex but we chose a model simplified and extremely flexible in order to provide decision-making aid so that a complete, and robust with an “acceptable” solution is not necessarily an optimal solution, but often a satisfactory solution obtained in a reasonable time [7].

For the manufacturer (plant manager, production manager and engineering manager) to accept an integrate solution, it is important to generate results that are familiar to them. Also, the 3S solution is a tool of decision-making aid which could help them to make the most suitable decisions quickly. Overall, the experiences from using this integration interface have been very positive, but they are not yet complete and final; therefore, we cannot summarize the benefit of this approach other testing complex examples.

Acknowledgment

This work is within the framework of an industrial project named “3.5 millions of Chertal” in collaboration between Mathematics and Operational Research department of the Mons Engineering Faculty and Research & Development division of the company Arcelor / Cockerill-Sambre. This study required three years of work.

References


APPLICATION OF ARTIFICIAL INTELLIGENCE ALGORITHMS FOR THE MODELING OF THE SURFACES OF ABRASIVE GRAINS

Piotr Stępień
Błażej Balasz
Tomasz Szatkiewicz
Tomasz Królikowski
Technical University of Koszalin
Department of Fine Mechanics
Ul. Raclawicka 15-17 75-620 Koszalin, Poland
E-mail: piotr.stepien@tu.koszalin.pl

ABSTRACT

The models developed reveal features which enable a generation of the surface of grains with properties statistically compliant with specified types of grains from different abrasive materials. The models of abrasive grains designed underwent an empirical verification. Due to the fact that the basic features which have an influence on the nature of the work of the grain (type of contact) are the parameters of the abrasive tool point, a comparative analysis was conducted in the range of checking the compatibility of the apex angle of the tool point $2\varepsilon$, the radius of the nose radius $\rho$ of the model grains and the proportion of the height $h_w$ of the grain to the width of its base $b$, as regards the real grains. The verification process consisted in determining the geometrical parameters of the models of grains generated. The values of the apex angle $2\varepsilon$ and of the rounding angle of the vertex $\rho$ were determined for various penetration depths of the tool point. The verification of the shape of the grains served to determine boundary values of the coefficients of the shape for individual types of grains, owing to which geometrical correctness of the modeled grains is ensured in the duration of the simulation process.

KEYWORDS: grinding, surface, modeling, simulations

INTRODUCTION

The efficiency and quality of abrasive machining processes has a decisive influence on the costs and quality of elements produced as well as whole products. The machining potential of abrasive tools is used insufficiently. One of more important reasons for an insufficient use of the machining potential is a slow development of new abrasive tools – development work focuses more on the improvement of the known technologies and not so much on the creation of new abrasive tools. Also, due to high costs of research into tools from ultra-hard materials concerning new tools, such research has not made a sufficient progress.

The use of the machining potential of tools depends of the optimization of the loading of abrasive grains, while typical empirical research allows solely for the designation of the global features of the process and not local ones, and temporary working conditions of active abrasive grains. The development of new modeling methods and the simulation of generation processes will facilitate a substantial progress in the creation of the basis of the system under development [4, 5, 6] and additionally it will enable to set assumptions for the creation of new abrasive tools with parameters to facilitate obtaining the expected results of machining, an increase of the productivity of the process and a much better use of the machining potential of grinding wheels.

MODELING OF THE ABRASIVE GRAIN SURFACE

Abrasive grains applied in machining can be divided into monocrystal, polycrystal and aggregate ones. The grain’s geometrical parameters play a vital part on the machining process as it is its shape that micro-machining processes are dependent on. Precise defining of the grain’s shape is very difficult due to a great variety of geometrical forms of grains created in the generating process. By a mutual agreement, abrasive grains are divided into groups defined as isometric, plate-like, pillar-like, swordtail-like and needle-like. There are numerous methods to assess the grain’s shape in such a manner, so that it should be possible, apart from geometrical features, to additionally assess in an indirect manner other features of grains, such as bulk density, abrasive ability or mechanical strength.

In order to make a complete assessment of an abrasive grain one should also determine the number and parameters of abrasive tool points located on the grain’s surface. This assessment is conducted through the measurement of the corner radius of tool point $\rho$, as well as of apex angle $2\varepsilon$ which determines the grain’s sharpness. The nose radius of the grain has a substantial influence on the machining process. Its size is closely related with the apex angle of the tool point, but the values of the radius for the same apex angles differ depending of the type of the abrasive material. Its change occurs also a result of the wear of the grains following the contact in the machining zone. It increases when there occurs a wear of the grain’s vertex, and it decreases when the fragments of the abrasive grain break up. While making an assessment of the usable features of grains one should also consider the structure of their surfaces (the surface morphology). Due to the fact that the penetration of a single abrasive grain in the material machined does not constitute more than 5% of its largest size [1], an important part is played by the features of the surface morphology of the grain, such as micro- and macro-cracks, notches in the surface, the number of vertices and their location (Fig. 1). All these factors play an influence on the nature of the grain’s work during the machining process, as well as its wear and ability to self-sharpen.
Figures 1: Pictures of abrasive grains taken with the use of a scanning microscope: a) monocrystalline Al2O3 b) green SiC, c) diamond, d) diamond covered with copper [1]

Analyses of the stereometry of real grains on the basis of research results quoted in literature [2, 3] formed the basis for the development of models of abrasive grains. In the simulation method developed it was assumed that what is important for the machining process is the grains’ contours protruding over the surface of the grinding wheel as well as their shape and size above the level of the binding material, as it is only those fragments of the grain that have an influence on the grain’s contact with the material and its wear. For this reason, the models develop describe only the stereometry of the part of the grain located above the geometrical surface of the grinding wheel. It was assumed in the modeling of the grains that the shape of the grain is described on a convex solid, with the local concavities of the grain’s surface being taken into account and modeled in the form of micro-roughness on the surface. In the model developed, the grain’s surface is described by a function whose components determine the grain’s shape \( f_{\text{kzr}}(x,y) \) and its micro-topography (irregularities of the shape) \( f_{\text{mp}}(x,y) \). The components of the function are combined in an additive or a multiplicative manner (1).

\[
Z_k(x,y) = f_{\text{kzr}}(x,y) + f_{\text{mp}}(x,y)
\]  

A numerical notation of the shape of the grain obtained is done with the use of a matrix of real numbers \( Z_k \) (2), whose size \([m, n]\) is determined on the basis of assumptions concerning the size of the grain modeled. The size of the matrix increases together with the growth of the grain’s sizes.

\[
Z_k(x,y) = \begin{bmatrix}
z_{11} & z_{12} & \ldots & z_{1n} \\
z_{21} & z_{22} & \ldots & z_{2n} \\
\vdots & \vdots & \ddots & \vdots \\
z_{m1} & z_{m2} & \ldots & z_{mn}
\end{bmatrix}
\]

where:

\[
z_0(x_1, y_1) = f_{\text{kzr}}(x_1, y_1) + f_{\text{mp}}(x_1, y_1)
\]

A numerical notation of the shape of the grain’s topography facilitates a modification of its shape during the simulation process caused by the grain’s contacts with the material machined, and also as a result of the dressing process of the grinding wheel.

The further part of the article presents the application of an elastic neuron network for the modeling of the surfaces of abrasive grains.

**MODELING OF THE SURFACE OF ABRASIVE GRAINS WITH USE OF AN ELASTIC NEURAL NETWORK**

In the neuron model developed, the output parameters are the number of the grain vertices, the apex angle and the vertex radius. As a result of the work of the system, a random model of a grain with set parameters is obtained. In the network developed, the weights of individual neurons represent the coordinates of points on the surface of the grain generated. The work of this neuron network consists in the change of the values of neuron weights, as a result of which the coordinates of points describing the surface of the modeled grain are obtained.

The proposed elastic neuron network consists of \( N \) neurons,

\[
A = \{n_1, n_2, \ldots, n_N\}
\]

where each one of them has a vector of weights assigned

\[
w_n \in \mathbb{R}^N
\]

to determine its location in the space of possible states \( \mathbb{R}^N \). Between the neurons in the network, there exists a system of elastic connections

\[
C \subset A \times A
\]

These connections are symmetric.

\[
c(i, j) = c(j, i)
\]

For each neuron, a set of neurons is assigned with which it is directly connected, also called adjacent neurons.

\[
N_k = \{i \in A | (c, i) \in C\}
\]

Each connection \( c(i, j) \) is assigned function \( f_c(d) \), called the function of elasticity. This function depends on the distances of weight vectors of connected neurons \( n_i \) and \( n_j \):

\[
d(n_i, n_j) = |w_i - w_j|
\]

in accordance with the agreed space metric \( \mathbb{R}^N \). Function \( f_c(d) \) is most often of a linear nature and is the same for all
the connections \( c(i, j) \in C \) (if the network is to be homogeneous). The value of function \( f_c(d) \) constitutes the quantity of the attractive force occurring between two adjacent neurons.

The network described, after its initiation, has the form of a rectangular grid, and so each neuron initially has 4 neighbors, with the exception of utmost neurons, which possess 2 or 3 neighbors each. At the same time, in this specific application for the simulation of the abrasive grain’s surfaces, the weights of utmost neurons are blocked. It means they do not change during the adaptation process.

The system of \( M \) nodes constituting characteristic points on the grain’s surface constitutes the output data for the network:

\[
L = \{ l_1, l_2, \ldots, l_N \} \in \mathbb{R}^N
\]

(9)

where each one of them has a vector of weights assigned to them:

\[
w_m \in \mathbb{R}^N
\]

(10)

to determine its location in space \( \mathbb{R}^N \). These nodes, in the case in question, constitute a system of characteristic points of the surface of the simulated abrasive grain.

During the network’s adaptation process (Fig. 2), the weight vectors of individual nodes affect simultaneously all the neurons located in the neighborhood determined by a certain radius. With the progress of the network’s adaptation both the neighborhood radius and the impact factor are subject to a reduction to lead to the network’s stabilization.

The purpose of the network’s adaptation in the case in question is to obtain such a final form of the network, i.e. such vectors of the weights of neurons \( w_m \) and such vectors of connections \( N \) that it should map the abrasive grain’s surface (Fig. 2d).

Two types of forces act on individual neurons: an attractive force from adjacent neurons and a force from the nodes, i.e. from the input data fed to the input of the network. For this reason, the following rule of the changes of the weights of neurons can be derived:

\[
\forall c(n,j) \neq 0 \left[ \Delta w_n = \beta (\sum_m \Lambda^m(n) + f_c(\kappa \sum (w_n - w_j))) \right]
\]

(11)

where:

- \( c(n,j) \) – connection between neurons \( n \) and \( j \),
- \( \beta \) – coefficient of network’s learning,
- \( f_c() \) – the elasticity function accepted,
- \( \kappa \) – elasticity coefficient, variable in the duration of the network’s adaptation process, proportional to the network’s temperature \( \beta \),
- \( \Lambda^m(n) \) – coefficient of an impact of node \( m \) on neuron \( n \) expressed with the following formula:

\[
\Lambda^m(n) = \frac{\exp\left(-\frac{|w_m - w_n|^2}{2\sigma^2}\right)}{\sum_m \exp\left(-\frac{|w_m - w_n|^2}{2\sigma^2}\right)}
\]

(12)

where: \( \sigma \) – effective range of the impact of nodes on the neurons.

In equation (11), the first expression is the force attracting every neuron \( n \) in the direction of the node (a characteristic point on the grain’s surface) \( m \) with the coefficient of impact \( \Lambda^m(n) \). The second expression is the total elasticity force which attracts every neuron in the direction of the adjacent neurons. The whole expression depends of the parameter of learning coefficient \( \beta \).

Figures 2: Individual stages of the adaptation process of an elastic neuron network: a) initiation of network in a grid form, b-c) adaptation of network, d) network’s final form depicting the surface of the simulated abrasive grain

As it can be seen from Fig. 2, the network in its first phase, right after its initiation (Fig. 2a) gradually maps the space of input signals. The elastic impact simulated results in the fact that the network during an expansion behaves like an elastic membrane and evolves like an equipotential surface in a certain vector field. The effect of the network’s work is a random surface of the abrasive grain with set parameters, which is then transformed into matrix Zk (2), used in the system of the simulation of the machining process.

Fig. 3. Sample final surface of simulated abrasive grains with crystalic edges marked
A generation of the grain’s surface may also occur with the use of different methods – as a process:

- of an additive and multiplicative accumulation of the function of the description of the shape, its disruptions, wear and micro-geometry;
- of tearing off fragments from the grain in various places and in a selected size;
- of precipitation and dispersion of precipitated particles in accordance with a certain procedure of the selection of local intensities of these procedures;
- of an automatic selection of the grain’s surface from a numerous set of grains previously recorded in the base (real ones or statistically compliant with them), with modifications correct in the range of a statistical compliance;
- of hybrid methods.

SI methods can be applied for each of the abovementioned methods.

REFERENCES

Woźniak K. 1982.: Podstawy procesu wytwarzania ziaren szlifujących w aspekcie regulowania ich kształtu, WSIhZ Koszalin.
Woźniak K. 1997.: Wpływ geometrii i wytrzymałości mechanicznej ziarna ścieżnego na jego zdolność ścierną, Zeszyty Naukowe WSIhZ Koszalin, nr 2, s. 257-270.
Balasz B., Królikowski T., Kacalak W. 2001.: Method of Complex Simulation of Grinding Process. Third International Conference on Metal Cutting and High Speed Machining Metz, France
SIMULATION OF THE FORMATION PROCESS OF REGULAR GROOVES ON SURFACE GROUND

Piotr STEPIEN, Blazej BALASZ
Technical University of Koszalin
Department of Fine Mechanics
75-620 ul. Raclawicka 15-17 Koszalin, Poland
e-mail: Blazej.Balasz@tu.koszalin.pl

ABSTRACT

This paper presents model and simulation of the methods of forming a regular geometrical structure of the surface in grinding with the circumference of a grinding wheel with deep screw grooves. The objectives of the simulation examinations of the machining process were as follows: an analysis of the formation process of the surface machined as well as an assessment of the level of regularity of the location and sizes of grooves cut on the object; an analysis of the distribution of sizes of the layers machined with the vertices of active grains, and an analysis of the load of the vertices of the grains and of the whole grinding wheel. The models of grinding and grinding wheel were presented, and results of simulation studies.

KEYWORDS: grinding, surface, modeling, simulations

1. INTRODUCTION

Technological surfaces, on which regularly distributed grooves have been formed: pits with proper shapes, have many properties useful for their exploitation [1 – 7]. It is especially recommended to form such surfaces on elements which work in conditions of sliding friction, where one can make use of the advantageous properties of such surfaces: a uniform distribution of the lubricating agent, a decrease of the friction factor through the creation of the so-called lubricity microclins, “absorption” of impurities and wear products in the areas of grooves, as well as good dampening properties.

One of the methods [8] of forming a regular geometrical structure of the surface consists in grinding with the circumference of a grinding wheel with deep screw grooves (Fig. 1). In the case of classical grinding wheels with a vitrified bond, screw grooves are cut with a single-grain diamond wheel dresser, while applying appropriately large dressing leaps s. The basic condition is that depth h of screw grooves be much greater than depth g of grinding. In practice, the parameter values of h = 0.05mm; s = 2mm; g = 0.02mm are applied.

Condition g < h results in the fact that only a part of vertices of abrasive grains are active, those which are located on a radius larger than R – g. The grinding wheel works with its whole height at the same time, and velocity vp of longitudinal feed is several times higher than in the case of standard grinding and obtains values of 0.5m/s. Owing to these two factors, the efficiency of the method is exceptionally high. It is also advantageous that one can in this manner machine materials with practically unlimited hardmesses, including ceramic materials [9].

Figure 1. Scheme of forming a regular geometrical structure of the surface

Depending on the number of screw grooves on the circumference of the grinding wheel and the manner of grinding, three types of grooved surfaces can be obtained (Table 1).

Table 1. Three types of grooved surfaces (view from above) with the use of a grinding wheels with screw grooves on the circumference; lighter areas are groove zones, and darker areas are the nominal (carrying) surface of the object.

Many experimental tests have proved an exceptionally high regularity of the distribution of grooves on the object’s surface. The shape of these grooves was also very repeatable. This is owing to the fact that each groove is cut with a group of numerous abrasive grains, and the wear of single grains does not have a significant impact on the change and location of the grooves.

An analysis of the formation process of groove surfaces is most comfortable to conduct for surfaces of type I, which is formed by a grinding wheel with a single screw groove, in one feed of the object. A theoretical description [8] of the formation process of such a surface was developed with the
assumption that all the vertices of abrasive grains are located on the nominal surface of the grinding wheel, and they are distributed perfectly uniformly. In spite of these simplifications, the description of the topography of the groove surfaces obtained is very accurate. However, it did not give the possibility to assess the level of differentiation of the load of active grains, and thus the possibility to model the wear process of the grinding wheel. The known tests and descriptions of this issue concerning the classic variations of grinding are not applicable in the case of this machining method. A good manner to describe this complex process was a computer simulation.

The objectives of the simulation examinations of the machining process were as follows:

- an analysis of the formation process of the surface machined as well as an assessment of the level of regularity of the location and sizes of grooves cut on the object;
- an analysis of the distribution of sizes of the layers machined with the vertices of active grains,
- an analysis of the load of the vertices of the grains and of the whole grinding wheel.

2. GRINDING WHEEL MODEL

The simulation tests concerned the case of machining of a groove surface of type I. The grinding wheel was conventionally divided into nw elementary layers with the same heights $\delta H$, being the result of the division of the fragment of height $\Delta H$ of the grinding wheel by the planes perpendicular to the axis of rotation (Fig. 2). In a given moment, several layers come in contact with the object machined. The knowledge of the load of a single and arbitrarily selected elementary layer during one turn of the grinding wheel makes it possible to sum in time the value of the loads of several neighbouring elementary layers, and in this way to determine the load of the whole grinding wheel. The locations of the grain vertices on the circumference of the elementary layer was determined in the polar system with angle $\alpha$ and radius $\rho$. The angular location of a given layer, being the result of the rotary motion of the grinding wheel, was determined with angle $\varphi$.

![Figure 2. Fragment of height $\Delta H$ of the grinding wheel divided into $n_w$ elementary layers with heights of $\delta H$](image)

It was accepted in the model that the screw groove on the circumference of the grinding wheel constitutes a representation of the vertex of the grinding wheel dresser, which can be described with a ball of radius $r_D$. The diagram of the nominal outline of the grinding wheel surface in the axial section is presented in Fig. 3.

![Figure 2. Fragment diagram of determination of a nominal outline of axial section $\rho_N(\varphi)$ of the grinding wheel, formed as a result of a representation of the vertex of the grinding wheel dresser ended with rounding radius $r_D$](image)

Having feed $s$ of screw line, radius $r_D$ of the rounding of the grinding wheel dresser and depth $h$ groove, length $l$ of the cylindrical part of the grinding wheel was determined, using the relation between the sides of ABC triangle. The equation of the circle of the outline of the circular part of the section of the vertex of the grinding wheel dresser was used in order to determine the coordinates of the outline of the cross section of the grinding wheel, in accordance with Fig. 3. Finally, the nominal outline $\rho_N(\varphi)$ of the axial section of the grinding wheel in the co-ordinate system, connected with the grinding wheel axis (Fig. 3) was determined in the following manner:

$$
\begin{align*}
\rho_N(\varphi) &= R, & & 0 \leq \varphi \leq l \\
\rho_N(\varphi) &= R - y(\varphi), & & l < \varphi \leq s
\end{align*}
$$

Going to the polar system, axial coordinate $z$ was exchanged for angular coordinate $\alpha$ in a simple linear transformation:

$$
\alpha = 2\pi \frac{z}{s}, \quad z = \alpha \frac{s}{2\pi}
$$

Changing coordinate $z$ for $\alpha$:

$$
y(\alpha) = \sqrt{r_D^2 - \left(\frac{s}{2\pi} - \frac{s+1}{2}\right)^2} - r_D + h
$$

Polar equation $\rho_N(\alpha)$ was determined of the nominal outline of the grinding wheel surface in a section perpendicular to the axis:

$$
\begin{align*}
\rho_N(\alpha) &= R, & & 0 < \alpha \leq \alpha_\varphi \\
\rho_N(\alpha) &= R - y(\alpha), & & \alpha_\varphi < \alpha \leq 2\pi
\end{align*}
$$

Circumference $2\pi R$ of the grinding wheel was divided into $n$ equal parts, while determining further angles $\alpha_i (i = 1 \ldots n)$ with step $\Delta \alpha = \frac{2\pi}{n}$. For each section, the nominal grinding wheel radius $\rho_N(\alpha)$ was calculated and described with formula (1).

The randomness of the distribution of grain vertices required the development of a probabilistic model, in which the location of individual grain vertices are subject to a chance. The location of the vertices of surface grains is usually determined in relation to the nominal grinding wheel surface, for example on a co-ordinate system presented in Fig. 4. Such a co-ordinate system was also accepted in the simulation model.
The arrangement of grain vertices in the circumferential direction $x$, and axial direction $z$, can be described relatively easily. It is assumed in compliance with [10-15] that grain vertices in these directions are arranged uniformly. This means that if one knows the average number of grain vertices on a unit of the grinding wheel surface, one can easily determine the average distance $l_x$ between them:

$$l_x = \frac{1}{\sqrt{c_{ax}}}$$  \hspace{1cm} (2)

Knowing $l_x$ is sufficient to generate coordinates $x$ and $z$ of grain vertices on any fragment of a grinding wheel surface of sizes $A*B$. It is enough to calculate the expected number $E(w)$ of such vertices:

$$E(w) = \frac{A \cdot B}{l_x^2} = A \cdot B \cdot c_{ax}^{-1}$$

and to generate $E(w)$ of values of circumferential $x$ and axial $z$ coordinates, using the uniform distribution for intervals $[0; A)$ and $[0; B)$. An example of the results of such an arrangement of grain vertices on the grinding wheel surface was presented in Fig. 5.

![Diagram of the co-ordinate system accepted in order to determine the location of grain vertices in relation to the nominal grinding wheel surface.](image)

**Figure 4.** Diagram of the co-ordinate system accepted in order to determine the location of grain vertices in relation to the nominal grinding wheel surface.

If it is to be further assumed that grain vertices are geometrical points on surface $0x\Omega$, then the probability of a grain vertex occurring in any direction of surface $0x\Omega$ is extremely small, and thus the distance between neighboring vertices in any direction approaches infinity. Such an approach practically excludes the possibility to generate grain vertices in any CPS direction, including circumferential direction $x$. However, it is worth noting that the impact of grain vertices on the material machined does not result in the point contact, but only occurs in a certain zone of non-zero sizes. Assuming the width of this zone as equal to $\delta H > 0$ (Fig. 5) the following reasoning can be conducted:

- on section of B width one can distinguish $B / \delta H$ of bands with $\delta H$ heights and $A$ lengths,
- the expected number of grain vertices in each band equals
  $$E(w) = A \cdot c_{ax}^{-1} \cdot \delta H$$  \hspace{1cm} (3)
- average directional distance between grain vertices in one band equals
  Accepting for example $c_{ax}^{-1} = 6/mm^2$ [10] and $\delta H = 0.1mm$, $l_x = 1.6(6)mm$ was obtained. The difference between $l_x = 0.408mm$ and $l_x = 1.6(6)mm$ is very large, however it should be remembered that distance $l_x$ constitutes the average distance between neighboring grain vertices measured in any direction, while $l_x$ is the average distance between neighboring grains in a single band of width $\delta H$. The expected number of grain vertices in a band of width $\delta H$ and length $2\pi R$ equals $E(w) = \frac{2\pi R \cdot c_{ax}^{-1}}{\delta H}$.
  For $R = 100mm$, $c_{ax}^{-1} = 6/mm^2$ and $\delta H = 0.1mm$, value $E(w) = 378$

Knowing average directional distance of grain vertices one can also assume that these occur in directions $x$ in a manner compliant with the Poisson’s process. With this assumption, the distance between further grain vertices is subject to the exponential distribution laws. Cumulative distribution function $F(\Delta x)$ of the exponential distribution makes it possible to generate values $\Delta x$ with the so-called reversion method of the cumulative distribution function. This requires the following transformations:

$$F(\Delta x) = 1 - \exp(-\lambda \cdot \Delta x) \quad \Rightarrow \quad \Delta x = -\frac{\ln(1 - F(\Delta x))}{\lambda}$$

Parameter of scale $\lambda$ is the converse of $l_x$. In the examinations, the following range of values $l_x = 0.83(3) + 3.3(3)mm$ was applied. By introducing in place of $F(\Delta x)$ any random number $RND$ from interval $[0; 1)$, one can calculate values $\Delta x$ from the following formula:

$$\Delta x = \frac{-l_x \cdot \ln(1 - RND)}{\lambda} = \frac{\Delta x \cdot R}{\lambda}$$  \hspace{1cm} (4)

To sum up, from zero to $2\pi R$, subsequently generated values $\Delta x$, a sequence of circumferential coordinates $x$, is obtained. Values $\Delta x$ were replaced with angles $\Delta \phi$, which makes it possible to determine the location of further grain vertices in the polar system.

An important assumption was accepted, i.e. that the arrangement of surface grain vertices in the radial direction is independent from their arrangement in the circumferential direction. The locations of surface grain vertices in the radial direction can be described with coordinates $\phi$, measured from the nominal grinding wheel surface in the centripetal direction (Fig. 4). Further, it was assumed that coordinate values $\phi$ are subject to Weibull’s distribution laws [11], determined with two parameters $m$ and $u$. Cumulative distribution function equation $F(\phi)$ of this distribution makes it possible to generate values $\Delta \phi$ with the use of the reversion method of the cumulative distribution function. This requires the following transformations to be made:

$$F(\phi) = 1 - \exp\left[-\left(\frac{\phi}{u}\right)^m\right] \quad \Rightarrow \quad \Delta \phi = -u \cdot \frac{\ln(1 - F(\phi))}{m}$$

where: $m > 0$ – shape parameter (abstract number),
$u > 0$ – scale parameter (expressed in length units).

Introducing in place of $F(\Delta y)$ any random number $RND$ from interval $[0; 1)$ makes it possible to calculate values $\Delta y$, from the following formula:

$$\Delta y = -u \cdot q \ln(1 - RND)$$

The selection of parameters $m$ and $u$ is of a great importance for the radial arrangement of grain vertices and consequently for their load during machining. In the examinations, the following values were applied: $m = 1.1$ and $u = 0.013\text{mm}$, more than a half of surface grain vertices is located on the depth up to 10$\mu$m. The remaining grain vertices are located deeper, and some of them are even on the level of 60$\mu$m below the nominal grinding wheel surface.

Figure. 6. Example of distribution of 500 grain vertices generated on the grinding wheel circumference and a histogram of their radial coordinates $\Delta y$ – the screw groove outline was not taken into account

Values of coordinates $\Delta y$ (5) were subtracted from nominal radii $r_p(\alpha)$ (1) of the grinding wheel and in this manner, radii $\rho(\alpha)$ of surface grain vertices were determined. If the generated radii $\rho(\alpha)$ of vertices met condition $\rho(\alpha) = R - g$, they became potentially active vertices, which can in favorable conditions come into contact with the material machined.

The manner of grinding wheel modeling described above was applied in the first module of the simulation program. The results of the calculations included the following:

- number of surface vertices on the grinding wheel circumference,
- number $w_{PA}$ of potentially active vertices,
- tables $(t = 1 \ldots w)$: $\alpha_i, r_p(\alpha_i), \rho(\alpha_i)$.

For a given set of input data: $R; s; h; r_p; g; C_{mH}; \delta H; m;$ $u$ sets of output values: $w; w_{PA}; \alpha_i; r_p(\alpha_i); \rho(\alpha_i)$ were obtained each time, which were practically unique. An example of the results obtained was presented in Fig. 7, presenting the location of vertices on diagrams in the polar system and in the axial section of the grinding wheel. The proportion between number $w = 279$ of surface vertices and number $w_{PA} = 126$ of potentially active vertices is worth noting.

Considering grinding depth $g$ and sense of angle $\varphi$ of grinding wheel turn (Fig. 2), one can indicate the grain vertex (marked with an asterisk), which will be the first one to contact the material machined. This characteristic grain vertex will start the first machining phase. The locations of further vertices working in phase 1 will be determined by the decreasing angles $\alpha$, until value $\alpha = 2\pi/2$ is reached.

For vertices working in phase 2, and located on the cylindrical part of the grinding wheel circumference, angles $\alpha$ decrease until the value of $\alpha = 0$ is reached. In the simulation, also potentially active vertices should be taken into account, which are located in the range of near angles $2\pi$. These vertices will work in phase 3. On the diagrams (Fig. 7), four ranges were marked of the vertices of grains working in the subsequent phases of the grinding wheel work.

Figure. 7. Example of a random arrangement of the surface grain vertices of the grinding wheel with a screw groove with the depth of $h = 0.06\text{mm}$ and feed $s = 2.3\text{mm}$; four ranges were marked of the vertices of grains working in subsequent grinding phases

3. Grinding Model

The path of the movement of grain vertices in relation to the object machined has the form of a cycloid, which is the result of putting together the rotary motion of the grinding wheel and the linear motion of the object. Designating by $r = R \psi$ the radius of the imaginary generating circle, the equations of cycloids (in the coordinate system connected with the object – Fig. 2) had the following form (for up cut grinding):

$$\begin{align*}
x &= r \cdot \varphi \cdot \rho (\alpha) \cdot \sin (\alpha + \varphi) \\
y &= -\rho (\alpha) \cdot \cos (\alpha + \varphi) + g - R
\end{align*}$$
\[ \varphi_{we} = \arccos \left( \frac{g - R}{\rho(a)} \right) - \alpha \]  
\[ \varphi_{wy} = 2\pi - \arccos \left( \frac{g - R}{\rho(a)} \right) - \alpha \]  
\[ (7) \]

Substituting \( v = 0 \) one can calculate (7) angles of input \( \varphi_{we} \) and output \( \varphi_{wy} \) of potentially active grains from the material machined. Knowing values of angles \( \varphi_{wy} \) of the output, they were substituted in formula (6) and coordinates of \( x_{wy} \) points were determined, in which these vertices leave the material. The first potentially active vertex is always active, since the material machined was not previously removed by other grains. To identify the remaining active vertices, the diagram from Fig. 8 was used, from which it is evident that the condition for the activity of a potentially active vertex is that its coordinate \( x_{wy} \) is larger than the coordinate of the exit of the preceding active vertex.

![Figure 8. Three possible situations in which next (n) potential active vertex could be active](image)

Having a list of active vertices, the minimal value of abscissa \( \{x_{we}\}_{MN} \) was determined, which determined the start of the groove cut on the object during one turn of the grinding wheel. The maximum value of abscissa \( \{x_{wy}\}_{MAX} \) determined the end of the groove. The difference of \( \{x_{wy}\}_{MAX} - \{x_{we}\}_{MIN} \) equals length \( l \) of the groove cut on the object. The groove length \( l \) was divided into 2,500 parts, and in this manner step \( \delta x \) of calculation conducted was determined:

\[ \delta x = \frac{\{x_{wy}\}_{MAX} - \{x_{we}\}_{MIN}}{2500} \]  
\[ (8) \]

In most cases, length \( l \) of the groove is a dozen or so of millimeters and step \( \delta x \) was of a few micrometers.

Further calculations consisted in determining paths (6) of the movement of active vertices in relation to the flat object machined. The nominal surface of the object was described with vectors \( \mathbf{x} \) and \( \mathbf{y} \). Vector \( \mathbf{x} \) included 2501 values of abscissa \( x \) varying from \( x_{MIN} \) to \( x_{MAX} \) with step \( \delta x \). Vector \( \mathbf{y} \) included 2501 values of initial ordinates \( y_i \).

For each active vertex ordinates \( y_b \) of cycloid were determined, while limiting the calculations only to the range of the zone of the potential contact of the vertices with the material, in which \( y_b > 0 \). The longitudinal profile of a groove cut during one turn of the grinding wheel was described with vectors \( \mathbf{x} \) and \( \mathbf{y} \). Vector \( \mathbf{x} \) preserves the initial values, while vector \( \mathbf{y} \) includes values \( y_b \), which were replaced with ordinates \( y_b > 0 \), describing the motion paths of subsequent active vertices.

4. ANALYSIS OF THE FORMATION OF GROOVE OUTLINE

An example of a single simulation, for a base data set (\( R = 100 \text{mm}; s = 2 \text{mm}; r_D = 2 \text{mm}; m = 1.3; u = 0.01 \text{mm}; \dot{l}_v = 1.66(1) \text{mm}; g = 0.02 \text{mm}; v_f / v = 0.03(3) \) concerns grinding with a grinding wheel with generated on circumference \( w = 387 \) grain vertices, slightly more than the expected number of vertices \( E(w) = 2\pi R / l_p = 376.99 \). Among all \( w = 387 \) surface vertices, \( w_{PA} = 236 \) are potentially active vertices, and \( w_A = 83 \) are active vertices. The proportion \( a = w_A : w_{PA} = 35.17\% \) is especially interesting, which describes the activity of those vertices which had an opportunity of contact with the material machined. Among \( w_A = 83 \) active vertices, first \( w_A = 7 \) are the vertices of phase 1, and the last \( w_{A3} = 3 \) are the vertices of phase 3. There were \( w_{A2} = 73 \) active vertices in phase two.

Proper proportions \( a_1 = w_{A1} : w_A = 8.43\%; a_2 = w_{A2} : w_A = 87.95\% \) and \( a_3 = w_{A3} : w_A = 3.61\% \) describe the activity of vertices in individual grinding phases. Active vertices can be seen in Fig. 9.

![Figure 9. An example of arrangement of grain vertices as a result of a single simulation for the base set of input data](image)

An important element of the assessment of the results is also the shape and sizes of the groove cut on the object machined. The movement paths of active vertices are presented in Fig. 10.

![Figure 10. An example of the movement paths of vertices as a result of a single simulation for the base set of input data and the resulting longitudinal outline of the groove](image)

Knowing the resulting outline of the groove, length \( l_c = 16.455 \text{mm} \) and the maximum depth of the groove \( \delta_{MAX} = 0.0159 \text{mm} \) were calculated. Field \( S = 0.273054 \text{mm}^2 \) of the longitudinal groove section was calculated as a sum:

\[ S = \sum_{i=1}^{2501} y_i \cdot \delta x = \delta x \sum_{i=1}^{2501} y_i \]  
\[ (9) \]

An analysis of each single result of the simulation is interesting, however due to its exceptionality it may lead to conclusions which should not be generalized. More general conclusions were obtained only after conducting a larger number of simulations. A full list of output quantities for 50
5. ANALYSIS OF LOAD OF ACTIVE GRAIN

It was assumed that the measure of a grain load are the sizes of the layer machined with a given grain. The notion of “layer machined = undeformed chip” used here is conventionally accepted here. Such a layer is not always removed (machined) – it can be deformed plastically or elastically, which depends of many factor, among others from the shape of the grain vertex, the sizes of the layer, local friction conditions, the velocity of the relative movement and the properties of the material machined.

A comparison of the subsequent movement paths of active grains makes it possible to determine the parameters of layers machined with these vertices. This required remembering a few additional values for each active vertex. Elements of vector \( \mathbf{y}_{k} \), including ordinates \( y_{k}(x_{k}) \) of the movement path of a given active vertex was compared with the current content of vector \( \mathbf{y} \), including ordinates \( y(x) \) of the object’s outline after machining with the previous active vertex. In cases where there was relation \( y_{k}(x_{k}) < y(x_{k}) \), the following operations were executed in the programme:

I) differences \( y_{k}(x_{k}) - y(x_{k}) \) were calculated and stored as the thicknesses of layers machined \( a_{k(i)} \):

\[
a_{k(i)} = y_{k}(x_{k}) - y(x_{k})
\]

II) thicknesses of layers machined \( a_{k(i)} \) were summed starting from zero and the end sum values were stored as \( S_{k} \):

\[
S_{k} = \sum_{i=1}^{m} a_{k(i)}
\]

III) thicknesses of layers machined \( a_{k(i)} \) were compared with initial value \( a_{\text{max(k)}} = 0 \) and current values \( a_{\text{max(k)}} \) were replaced with the greater of values \( \{a_{k(i)}, a_{\text{max(k)}}\} \).

The manner to determine the parameters of the longitudinal section of the layers machined with subsequent active vertices is illustrated in the diagram from Fig. 12, which presents the movement paths of eight subsequent active vertices. The first three \((k = 1, 2, 3)\) active vertices (Fig. 12) start and finish work in points located on the object’s nominal surface \((y = 0)\). The fourth vertex \((k = 4)\) works on a shorter path and contacts the material machined only on the depth of ca. 0.012mm. The result outline of the object’s surface \( y(x_{k}) \), which is the effect of the contact of the vertices with nos. 1 – 4, was presented in Fig. 12 as a thick grey line.

While determining the load of the fifth vertex, its path \( y_{5}(x_{5}) \) was compared with the result outline \( y(x_{5}) \), which is the result of the work of the four previous active vertices. In the sections, in which \( y_{5}(x_{5}) > y(x_{5}) \), the difference was \( a_{k(5,i)} \) determined, obtaining at the output of vertex 5 the value of the sum stored as \( S_{5} \). The thicknesses of layer \( a_{k(5,i)} \) were marked in Fig. 12 in the form of two-way arrows.

Considering the small (several micrometers) values of step \( \Delta x \) on can fairly accurately calculate field \( \Delta S_{k} \) of the longitudinal section of a layer machined with \( k \)-th active vertex:

\[
\Delta S_{k} = S_{k} \cdot \Delta x
\]

Length \( t_{k} \) of the contact path of the \( k \)-th vertex with the material machined was calculated as a sum of elementary relocations (Fig. 12) of vertices \( \Delta r_{i} \):

\[
\Delta r_{i} = \sqrt{(x_{i} - x_{i-1})^{2} + (y_{i} - y_{i-1})^{2}} = \sqrt{\Delta x^{2} + (\Delta y)^{2}}
\]

The average thickness \( a_{\text{ave(k)}} \) of the layer machined with \( k \)-th vertex was determined with the following formula:

\[
a_{\text{ave(k)}} = \frac{\Delta S_{k}}{t_{k}}
\]

For each active vertex, also the maximal thicknesses \( a_{\text{max(k)}} \) of the layer machined were determined:

\[
a_{\text{max(k)}} = \text{MAX}\{a_{k(i)}\}
\]

Value \( a_{\text{max(5)}} \) for the fifth vertex was shown in Fig. 12 as a thick arrow.

The results obtained in 50 simulations for the base set were presented in Fig. 13. Parameters \( \tau, \Delta S_{k}, a_{\text{ave}} \), \( a_{\text{max}} \), which describe \( w_{d} = 4116 \) of layers machined, were expressed as the function of angle \( \alpha \), and were presented in the order concerning the contact of vertices with the material machined, as in Fig. 7.
Knowledge of the sizes of the layers machined made it possible to conduct further simulation examinations, in which the values were determined of the forces acting on individual grain vertices, and of the force acting on the whole grinding wheel. For this purpose, the diagram for Fig. 2 was used, and the results of the simulation of single elementary layers were integrated for the whole grinding wheel.

6. SUMMARY

1. The results obtained in subsequent simulations are diversified, however their changeability is small. Most of changeability coefficients do not exceed 10 per cent, that is the value usually accepted as the boundary value for the so-called random variables quasi-permanent.

2. Only numbers of vertices \( w_{41} \) and \( w_{42} \) working in phases 1 and 3, proved to be „statistically unstable”, which is confirmed by a great difference of the maximum and minimum values, as well as by the large width of the confidence interval of the average value. The results of this variability apply of course also activity coefficients \( a_1 \) and \( a_2 \).

3. The randomness of the arrangement of grain vertices is revealed with the comparison of the result outlines of grooves cut on the object’s surface and is visible (Fig. 11) especially in the outlines of the bottom of the grooves.

4. The bottom of the groove, formed with vertices of different radii \( \rho (\alpha) \) is not flat. It should be expected that the roughness level of the bottom of the grooves will be fairly high: the height of the unevenness of the groove bottom reached (for the base data) values of up to 0.005 mm.

5. Some vertices do not continue their penetration into the material formed directly by the previous vertex, but start their contact with the material in places left by vertices working much earlier (Fig. 10).

6. No significant differences were found as regards the location of the starts (\( x_{\text{MIN}} \)) and ends (\( x_{\text{MAX}} \)) of the grooves, nor as regards their length \( \ell \).

7. A random arrangement of grain vertices results in the fact that the real areas \( S \) of the longitudinal section of the grooves are substantially smaller than the theoretical one, as radius coordinates \( \delta_r \) of grain vertices are usually larger than zero. For this reason, the averaging outline of the grooves has a depth smaller that depth \( g \) of grinding (Fig. 11).

8. Grains working in phase 1 have a greater load than grains working in phase 2, and grains of phase 3 have the smaller load.

9. In accordance with theoretical predictions, grain vertices located on the boundary of phases 1 and 2 have the greatest load: this especially applies to (Fig. 13) length \( r \) and area \( \Delta S \) of the layers machined with these grains.

10. The randomness of the arrangement of grain vertices results in the fact that the dispersion of the parameters of the layer machined in relation to their nominal values is exceptionally high.

Simulations performed for different input data allowed for an assessment of the influence of the changes in the machining parameters on the process and grinding results in these specific conditions, being the result of the method applied (Fig. 1). The abovementioned conditions were also confirmed, which can generally be summarized in the statement that the randomness of the arrangement of grain vertices has no significant influence on the „external” machining results (the shape and location of the grooves on the object’s surface), but has a strong impact on the arrangement and values of the load of active vertices, i.e. on the „internal” features of the process.

7. REFERENCES


**BIOGRAPHY**

**BŁAŻEJ BAŁASZ** was born in Trzcianka, Poland and went to the Technical University of Koszalin, where he studied fine mechanics and information theory and obtained his degrees in 1994. In 2003 obtained Ph.D. degree in mechanical engineering for dissertation on simulation systems in grinding processes. He is interested in simulation systems of engineering processes, neural networks and information processing.

**Piotr Stępień, Ph.D.** Works for over 30 years as a tutor at Mechanical Department of Technical University of Koszalin. Main field of interest: abrasive technology, statistics, operational research, econometrics, project management. Author and co-author of 32 articles concerning grinding tools and technology.
ON THE SIMULATION OF ROUNDESS PROFILES OBTAINED BY TURNING

Giovanni MORONI  
Dipartimento di Mecanica  
Politecnico di MILANO  
giovanni.moroni@polimi.it

Massimo PACELLA and Alfredo ELIA  
Dipartimento di Ingegneria dell’Innovazione  
Università degli Studi di LECCE  
rassimo.pacella@unile.it

KEYWORDS  
Coordinate measuring machines, geometric tolerance, manufacturing signature.

ABSTRACT  
Quality of mechanical components is more and more often related to geometric tolerances, e.g., roundness, flatness, parallelism, runout and position. Usually, geometric tolerances are checked using a Coordinate Measuring Machine (CMM), which samples a set of points on a surface, and then, using appropriate fitting algorithms, estimates a tolerance. In planning the CMM based inspection of a mechanical component feature, cleverness may be added with the definition of an empirical model representing the measured surface (or profile). In this paper, two empirical models of roundness profile obtained by turning are described. Such models allow foreseeing the “signature” left by the turning process. By using such roundness models, the aim of the present paper is to exploit computer simulation in order to estimate the relationship between the out-of-roundness of turned items and two factors: 1) the number of measurement points taken, 2) the fitting algorithm implemented. The widest used techniques of geometric best fitting, i.e. least square and minimum zone methods, have been implemented in this work in order to estimate the out-of-roundness value for each simulated profile.

INTRODUCTION  
With reference to geometric (position and form), dimensional and roughness specifications, all mechanical workpieces exhibit departures from the ideal. As a matter of fact, macro- and micro-deviations from the ideal geometry can be related to the specific process used to machine workpieces and to many different causes such as 1) machine tool (vibrations in the machine-tool-workpiece system, travel deviations related to the guiding system, thermal instability, spindle errors); 2) fixturing system (position and orientation deviations of the workpiece from the nominal position in the clamping fixture); 3) process conditions (operating parameters like rotational speeds and feeds); 4) tool (geometry, wear state, friction conditions during material removal); 5) workpiece (actual geometry before machining, material’s properties).

Such variations, which are resulting from natural causes inherent in all manufacturing processes, can have significant effects on the function of the final assembly and should be within predetermined tolerance specifications.

The major aim of production metrology is to check whether the shape of a workpiece complies with its functional requirements translated by the designer into tolerances of dimensions, position (relative to a reference geometric element), form and roughness.

A coordinate measuring machine (CMM) is a computer-controlled device for assessment of manufactured products with external datum reference. A CMM can be used for measuring dimensions, form and positions in a single operation, with just one instrument based on a unique coordinate system. The conventional operation of CMM consists of two phases: 1) measurements of coordinates of points on the machined surface, and 2) estimation of the substitute geometric features of the workpiece.

In typical evaluations of geometric and form tolerances, the substitute geometric feature is constructed by best-fitting the measurement data. Geometric best fit can be viewed as a numerical transformation between the measurement data and their nominal geometry. Several best fit methods (i.e., cost functions) can be adopted. These methods include minimum zone (MZ) and least-squares (LS). MZ looks for a couple of geometrical nominal features at minimum distance that includes the whole set of measurement points. LS associates one substitute feature to measurement points, and calculates the maximum peak-to-valley distance of the measurement points from the substitute feature.

A closer analysis of the specific process involved in manufacturing the feature can add cleverness in tolerance estimation. This added cleverness can result in a reduction of inspection costs for a fixed level of accuracy. Cleverness coming from the knowledge of the manufacturing process is related to the definition of the empirical model representing the “signature” left by the process on the feature machined. This manufacturing signature is the systematic pattern that characterizes all the features machined with that process (Wilhelm et al., 2001). In fact, measured surfaces (or profiles) often present a systematic pattern and a superimposed random noise: the first is mainly due to the process used in specific operation conditions, while the second is due to unpredictable factors which are usually known as “natural variability”.

The present paper describes two methodologies to model process signature related with form features. The first methodology is based on classical Fourier analysis of machined surface while the second methodology is based on the auto regressive moving average with exogenous variables (ARMAX) model of appropriate orders for modelling the process signature. Since the measurement of roundness for manufactured parts is one of the most frequently used procedures in metrology, it is adopted as the reference nominal geometry in this work.

By using such roundness models, the aim of the present paper is to exploit computer simulation in order to estimate the relationship between the out-of-roundness of turned items and two factors: 1) the number of measurement points taken, 2) the fitting algorithm implemented.

An overview of this paper is as follows. In section 2, a review of reference literature is presented. Section 3 presents the two roundness models exploited in this work for simulations. Section 4 describes the simulation work and the related results. Finally conclusions are given in section 5.
STATE OF THE ART

The problem of process signature modeling has been receiving increasing attention in the last few years. In the case of circular profiles, a diffused approach was based on modeling radial deviations with a periodic function. In fact, different researchers have explored possibilities related to modeling roundness using Fourier series expansion. One of the earlier approaches was presented by Yeralan and Venturini (1988) which used a Fourier series model to investigate the statistical aspects of roundness measurement procedures. Similarly, Chang and Lin (1993) adopted the Fourier representation and combined it with computer simulation techniques to find a minimum set of data sufficient to quantify circularity errors.

Another approach based on Fourier analysis has been presented by Cho and Tu (2001). The authors proposed a statistical model which can allow simulating sampling of roundness profiles. Since Fourier transform of a periodic function (as radius variations of cylindrical surfaces) can be interpreted as a sum of sinusoids, the roundness profile is described in their model by the statistical distribution of Fourier harmonic amplitudes. In particular, the authors analyzed the case of cylindrical parts made by turning, and using experimental data, they identified a beta distribution for each of the harmonic amplitudes (where beta distribution parameters change harmonic from harmonic).

Such an approach can be useful to model a roundness profile, but only when data measured on that profile are not autocorrelated. Most of the times however data collected on a machined profile are autocorrelated because they are obtained in similar condition of the machining process and of the measurement system. Colosimo et al. (2004) pointed out such a problem, and in their work they proposed a robust and objective procedure for manufacturing signature modelling in which the empirical model is determined through a statistical criterion aimed at separating systematic behaviour (signature) from random noise. The proposed method was able to deal with autocorrelation, however, the large positive autocorrelation (about the fourth order) observed in the measured data exploited in their work, was only partially resolved by using an empirical gapping strategy (i.e. reducing the number of measured points by skipping some adjacent measurements). Subsequently, Colosimo and Pacella (2005) presented a novel method based on the auto regressive moving average with exogenous variables (ARMAX) model. The goals of that work were to present a methodology for identifying process’s signature when statistical autocorrelation affects the random noise, to compare different ARMAX models for the process’s signature modelling and to establish a procedure to select the best one.

In this research, the model proposed by Colosimo and Pacella (2005) and Cho and Tu (2001) are exploited for computer simulation. In the following section, the two models (labelled “ARMAX” and “Cho & Tu” respectively) are described in detail.

PROCESS’ SIGNATURE IN TURNED CIRCULAR PROFILES

Both of the approaches exploited in this work are based on modelling circular form errors by means of a series of harmonics superimposed on an ideal circle. The use of harmonics is justified because machined circular profiles tend to be periodic due to rotational error motions of machine tools. Supported by experiments, these roundness models are proved to be effective in simulating process’ signature in turned circular profiles.

ARMAX based model

A sampled profile can be modelled as a weighted sum of analytical components, plus a random error. Let \( t = 1, 2, \ldots, N \) represent the index of data points in the sampled profile, where \( N \) is the number of equally spaced points measured on that profile. For a given index \( t \), let \( Y_t \) be the t-th measurement on the sampled profile, the ARMAX based model of circular profile employed in this work is a specific instance of the following general model:

\[
Y_t = \hat{b}_0 + \sum_{k=1}^{K} [b_{2k-1} \cos(f_{k}t) + b_{2k} \sin(f_{k}t)] + \frac{1}{A(2^N)} \varepsilon_t
\]

where \( Y_t \) is the single output signal and represents the radius of the roundness profiles, measured at the angle \( \theta_t = (2\pi/N)(t-1) \) rad. \( \varepsilon_t \sim (0, \sigma^2) \) represents an univariate white noise, \( 2^N \) is the backward shift (or backshift) operator, i.e. \( 2^N x_t = x_{t-2^N} \), while:

\[
\frac{1}{A(2^N)} \varepsilon_t = \frac{1}{1 - a_1 2^{-1} - a_2 2^{-2} - \ldots - a_p 2^{-p}} \varepsilon_t
\]

represents the AR(p) (autoregressive) model assumed for the noise. The signature model is a linear combination of harmonic functions where each term \([b_{2k-1} \cos(f_{k}t) + b_{2k} \sin(f_{k}t)]\) represents the k-th harmonic, characterized by a frequency \( f_k = k \cdot 2\pi/N \) rad/sample, amplitude given by \( \sqrt{b_{2k-1}^2 + b_{2k}^2} \) and phase given by arctan \((b_{2k-1}/b_{2k})\). Each harmonic component can be considered as one of the exogenous variables of the ARMAX model.

The model of equation (2) has been fitted on actual roundness data obtained from a test case. In particular, CMM observations of roundness profiles, i.e. cylinders machined by turning, were exploited. The set of machined artefacts used as the source of data for this work has been described in detail elsewhere (Colosimo et al. 2004).

Figure 1 shows the amplitudes and the phases of the first 50 harmonics terms obtained after applying Discrete Fourier Transform (DFT) to all the 100 samples in the set of machined artefacts used as test case. It can be noticed that the null and the first harmonics (0 UPR and 1 UPR respectively) do not appear as significant (i.e., their associated amplitudes are zeros). This effect is related to pre-treating of data, which consisted in centring of workpiece and radius suppression. In fact, subtracting the least squares estimation of the radius results in cancelling the effect of the null harmonic, while re-centring the profile results in removing the first harmonic, i.e. \( b_0 = b_1 = b_2 = 0 \) (Cho and Tu 2001).

By looking to the diagrams in Figure 1 it clearly appears that the roundness signature is characterized by two main harmonic terms, namely the second (2 UPR) and the third one (3 UPR). Therefore, the process signature’s model in (1) can be rewritten as follows:

\[
\sqrt{N} \left[ b_3 \cos(f_2t) + b_4 \sin(f_2t) + b_5 \cos(f_3t) + b_6 \sin(f_3t) \right]
\]

where \( f_k = k \cdot 2\pi/N \) is the frequency of the k-th harmonic \((k = 2, 3)\) and the factor \( \sqrt{N} \) is due to the selection of normalized harmonic predictors, usually adopted in DFT theory.

Once process’ signature model has been identified, the order of the polynomial \( A(2N) \) is selected. With reference to the roundness data collected, we identified order \( p \) as the lower value inducing
uncorrelated residuals. In particular, according to our investigation of the real case, an AR(2) is the most suitable model. In fact, when the order of the model is increased, residuals whiteness does not improve significantly.

![Graph showing amplitude and phase for the first 50 harmonic terms of the 100 roundness profiles measured. Abscissa scale: Undulation Per Revolution (UPR); ordinate scale: amplitude.](image)

**Figure 1:** Amplitude (a) and phase (b) for the first 50 harmonic terms of the 100 roundness profiles measured. Abscissa scale: Undulation Per Revolution (UPR); ordinate scale: amplitude.

Parameters estimates were computed using the prediction error method implemented in MATLAB (System Identification Toolbox - Ljung 1997). With this estimated ARMAX model, the final step consisted in performing a diagnostic check on residuals obtained (residuals of the model should be white and not correlated to the exogenous variables of the model). Both the assumption of no autocorrelation of residuals and no cross-correlation between residuals and harmonic components were indeed verified. Considering the AR(2) identified for the noise and the signature model identified the general ARMAX model assumed for roundness data is given by:

\[ y = \frac{1}{1 - \alpha_1 b_1 - \alpha_2 b_2} b_2 a \sum_{k=1}^{50} (b_{2k-1} \cos(f_k t) + b_{2k} \sin(f_k t)) \]

The vector of six model parameters \( [b_1, b_2, \ldots, b_{20}, \alpha_1, \alpha_2] \) is stochastic as it can change from profile to profile. The best-fitting distribution function for such parameter vectors is a multivariate normal distribution with mean vector and variance-covariance matrix reported in tables 1 and 2.

Finally, the error \( \varepsilon \) is modelled as Gaussian white noise, of null mean and standard deviation equal to \( \sigma_\varepsilon = 0.374 \mu m \).

It is worth noticing that the model of equation 4 can be also considered a series with seasonal component and random component. The seasonal component is described by a Fourier series while the random component is described by an autoregressive structure. Combining a Fourier series to a Box-Jenkins autoregressive structure is a more parsimonious parameterization of the model for circular profiles.

<table>
<thead>
<tr>
<th>Table 1: ARMAX parameters (100 samples): mean vector estimate (mm)</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \hat{b}_1 )</td>
</tr>
<tr>
<td>-0.0341</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Table 2: ARMAX parameters (100 samples): covariance matrix estimate (mm²)</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \begin{bmatrix} 0.0004 &amp; -0.0002 &amp; 0.0001 &amp; 0 &amp; 0.0001 &amp; 0.0003 \ -0.0002 &amp; 0.0004 &amp; 0.0001 &amp; 0 &amp; 0.0001 &amp; 0.0003 \ 0.0001 &amp; 0.0001 &amp; 0.0002 &amp; 0 &amp; 0.0001 &amp; 0.0003 \ 0 &amp; 0 &amp; 0 &amp; 0 &amp; 0.0003 &amp; 0.0003 \ 0.0001 &amp; 0.0001 &amp; 0.0001 &amp; 0.0003 &amp; 0.0072 &amp; 0.0012 \ 0.0003 &amp; -0.0002 &amp; 0 &amp; 0.0003 &amp; 0.0012 &amp; 0.0036 \end{bmatrix} )</td>
</tr>
</tbody>
</table>

**Cho & Tu model**

In their paper, Cho and Tu (2001) proposed a DFT based representation and suggested to consider the first fifty harmonics to represent roundness signature obtained by turning. In our simulation model, centring of workpiece and radius suppression are both considered as the main interest is only on its roundness. Consequently, also in this case, the resulting frequency model does include neither the zero order harmonic nor the first order one. Such an approach implies that 98 parameter coefficients have to be identified; i.e. the amplitudes and phases of the harmonics from the 2nd to the 50th. The model can be hence described as follows:

\[ \mathbf{b}' \mathbf{u}_t = \sum_{k=1}^{50} (b_{2k-1} \cos(f_k t) + b_{2k} \sin(f_k t)) = \sum_{k=1}^{50} c_k \cos(f_k t - \phi_k) \]  \hspace{1cm} (5)

where \( c_k = \sqrt{b_{2k-1}^2 + b_{2k}^2} \) and \( \phi_k = \arctan(b_{2k-1}/b_{2k}) \).

The model of Cho and Tu (2001) is stochastic with parameters determined based on profiles scanned for parts produced by different machining processes. The authors demonstrated that the best fitting distribution function for amplitudes \( c_k \) is the beta distribution where distribution parameters change harmonic from harmonic (beta parameters can be found in table 3 of the cited article), while phase angles, \( \phi_k \) can be assumed uniformly distributed within 0 and 2π.

It should be noticed that the harmonic amplitudes, which appear in the model proposed by Cho and Tu (2001), are normalized in order to obtain a tolerance value ranging in [0, 1]. Hence, in the profile model implemented in our research, a tolerance multiplying factor for harmonic amplitudes (say \( c_k \)) from 2nd to 50th order has been introduced in order to obtain a tolerance value ranging in [0, 1].

Additionally, it should be noticed that such model does not distinguish signature from error \( \varepsilon \). In particular, in their work \( \varepsilon \) was considered related to high frequency components in the DFT, so the sum of harmonics should be divided in two parts, the first of which contains 50 harmonics constituting signature, while the remaining components constitute the noise.
Therefore, in this work a slight modification of the model proposed by Cho and Tu (2001) is implemented. The implemented model is described by the following equation (6):

\[ Y_i = \sum_{k=2}^{n} \frac{b_k}{\sqrt{2}} \cos(f_k \tau_1) \sin(f_k \tau_2) + \epsilon_i \]

\[ = \sum_{k=2}^{n} \frac{b_k}{\sqrt{2}} \cos(f_k \phi_1) + \epsilon_i, \]

where \( \epsilon_i \) is modelled as Gaussian white noise of standard deviation equal to \( \sigma = 0.374 \mu m \) and \( \tau_i = 0.035 \).

It is worth noticing that the model of equation (6) can be considered as a particular case of the general ARMAX model of equation (1) where \( K = 100, b_0 = b_1 = b_2 = 0, \) and \( p = 0 \), i.e. \( A(\phi) = 1 \).

**FITTING CRITERION AND SAMPLING STRATEGY**

Generally, in geometric and form tolerances assessment, a substitute feature is constructed by best-fitting the measurement data. Geometric best fit is a numerical transformation between measurement data and their nominal geometry.

In this work, the analysis was performed on computer to determine the parameters of the geometrical best fit. For circular profiles, one of such parameters is represented by roundness error (also called out-of-roundness - OOR). OOR is an important geometric characteristic as it affects the functionality of products in a number of ways.

In order to estimate the OOR value, both the minimum zone (MZ) and least-square (LS) method have been considered.

MZ looks for a couple of geometrical nominal features (e.g. a couple of concentric circles for roundness) at minimum distance that includes the whole set of measurement points. The MZ method has received much attention in recent years as it best conforms to the ISO standard for form tolerances. The method always consists of the minimum variation, given a set of measurement data. However, it is very sensitive to asperities.

Different mathematical or geometry-based techniques to estimate the couple of minimum-distance nominal features have been reported in the literature. In this work, the algorithm used to evaluate the MZ circularity was derived from the solution given by Carr and Ferreira (1995) for the roundness problem. The implemented MZ algorithm is based on a minor modification of such a method, in which it allows to identify decision of good search direction and search step length for linearization of the optimization problem. A program for the purpose was developed in MATLAB, tried out on several benchmark data sets and found to give correct results (Pacella et al. 2004).

LS method, on the other hand, is the most commonly used algorithm for form tolerance evaluation as it is based on sound mathematical principles that minimize the sum of squared deviations of measured points from the fitted feature. The LS method associates one substitute feature to measurement points (e.g. one circle for roundness), and calculates the maximum peak-to-valley distance of the measurement points from the substitute feature.

LS method requires that the sum of the squares errors must be minimized, hence all the measurement points contribute to the best fit result. Although the form error computed from the extreme points can be slightly higher than that obtained from the MZ method, the LS fitted feature is very stable and much less sensitive to the effects of asperities, making it suitable for many practical applications.

In this work, we do refer to the implementation developed by NPL (the UK’s National Physical Laboratory) and based on the Least Squares Geometric Elements (LSGE) library for MATLAB. The LSGE library consists of functions to find the least-squares fit of geometric shapes to data, implementing a number of geometric fitting routines key functions. It is based on a general purpose nonlinear least-squares solver that takes as input function-and-gradient routines, and these routines are implementations of the geometric evaluation key functions.

This library was also tried out on several benchmark data sets and found to give correct results (Pacella et al. 2004).

**ANALYSIS OF SIMULATION RESULTS AND DISCUSSION**

The number of sampled measurement points considered in this work spans from 6 to 748. In particular, the roundness models of equations (4) and (6) were used to create profiles with nominal diameter of \( \Omega 25.4 \text{ mm} \) and \( \Omega 26 \text{ mm} \) respectively by means of computer simulation. Each simulated profile, which can be considered the result of a turning machining, consists of a 748 points as the perfect operator for both the nominal diameters \( \Omega 25.4 \) and \( \Omega 26 \text{ mm} \) consists of 748 discrete measurement data per circle. These data were simulated as measurement points evenly distributed around the measured circle.

For each down-sampled profile the best fit parameter (i.e. the out-of-roundness) was collected and compared to the one obtained by 748 points.

Let \( \{1, \ldots, 10000\} \) be the index of simulated profiles.

\( L_{(748)} \) is the least squares form error estimation for the \( i \)-th profile obtained by the whole a set of 748 available measurements. \( M_{(748)} \) represents the equivalent estimation based on the minimum zone algorithm, which represents the true value of roundness as required by standard ISO. The error in approximating MZ value with the LS is measured as follows:

\[ \text{err}_{(748)} = \frac{L_{(748)} - M_{(748)}}{M_{(748)}} \]

where a value closed to zero imply a better accuracy.

Figure 2 depicts the box plots and range plots of 10000 replications for response \( \text{err} \) both for the ARMAX and Cho & Tu (2001) model: denoted by \( \text{err} \) (ARMAX) and err (Cho&Tu) respectively. From figure 2, it can be noticed that the LS algorithm overestimates the roundness error calculated by the MZ algorithm by approximately 6%.

![Figure 2: err simulation results (10000 replications).](image-url)
These results confirm that the OOR computed by LS algorithm are statistically higher than that obtained from the minimum zone method. Therefore, the choice of the algorithm influences the estimation in the sense that the MZ supplies an expected value lesser than that obtained from LS.

Seven additional sampling conditions were also produced by a regular down-sampling of 347, 187, 94, 47, 23, 12 and 6 equally distributed points from each of the original vector of 748 points, and the following error was calculated for each simulated profile.

\[
\text{err}_a(\text{points}) = \frac{LS(\text{points}) - MZ(748)}{MZ(748)}
\]

Figure 3 shows the mean plots (where the simulation error is negligible) as function of the number of points. It clearly results that the “accuracy” of the best fit result improves as the number of sampling point increases. However, it is worth noticing that increasing the number of sampling points can result in considerable increase of inspection costs in actual cases.

Interestingly, it can be noted that since the LS algorithm overestimates the real OOR value, this allows reducing the number scanned points (with the obvious advantage to cut off inspection costs).

For example, in the case of ARMAX model by measuring one half of the points (374) instead of the number required by the standards (748), and implementing the LS algorithm, the MZ OOR value is approximated with just a mean error of -0.160%. In the case of Cho and Tu (2001) model a number of scanned points which range between 94 and 187 can even allow obtaining a null mean error.

CONCLUSIONS AND FUTURE ISSUES

In designing the measuring strategy, cleverness may be added with the definition of an empirical model representing the signature left by the process on a machined feature.

In this paper, two empirical models of roundness profile obtained from turning are described. The relationships between the number of measurements and fitting algorithms are examined by computer simulation and benefits in terms of inspection costs reduction were demonstrated. The obtained results demonstrated that by using simulation model provides desirable insights into the advanced tolerance analysis of circular profiles.

Although applied to a specific process (turning) and to a specific tolerance (roundness), the simulation approach proposed in this paper is quite generic and can be easily extended to different signatures. Further advantages of signature analysis should be investigated in the near future. For instance, the use of different representations of the signature should be explored.

ACKNOWLEDGEMENT

The work described in the paper has been partially funded by the Ministry of Instruction, University and Research of Italy (MIUR)

REFERENCES


Ljung, L., 1997 System identification toolbox user’s guide (for use with MATLAB) © The Mathworks Inc.


ELECTRONICS DESIGN SIMULATION
Fuzzy Simulation to Speedup Computer Design

Giuseppe Ascia, Vincenzo Catania, Alessandro G. Di Nuovo, Maurizio Palesi, Davide Patti
Dipartimento di Ingegneria Informatica e delle Telecomunicazioni
Università degli Studi di Catania
Viale Andrea Doria 6, 95125 Catania, Italy
e-mail: {gascia,vcatania,adinnuovo,mpalesi,dpatti}@diit.unict.it

KEYWORDS
Fuzzy Simulation, Evolutionary Design, Simulation-based design.

ABSTRACT
This paper deals with application of fuzzy simulation in computer system design with a particular emphasis on the role of simulation in the overall design activity of an embedded system. Simulation occurs at many levels, from circuit to system, and at different degrees of detail as the design evolves. In this scenario the use of fuzzy simulation enables quick and accurate design decision in the early stages of computer design, at the processor and system levels. As a case study we present the design of a high performance low power embedded system targeted for multimedia applications. We show how our fuzzy based simulation approach is a good candidate to replace slowly traditional simulation approaches with a drastically reduction of the total design effort.

INTRODUCTION
The growing demand for embedded computing platforms, mobile systems, general-purpose handheld devices, and dedicated servers coupled with shrinking time-to-market windows are leading to new core based system-on-a-chip (SoC) architectures, and to design methodologies which stress design reuse by exploiting a new design paradigm known as Platform Based Design (PBD) (Chang et al. 1999). When a PBD methodology is used, the platform is the pillar around which the overall design activity focuses on.

Often a platform consists of a parameterized microprocessor, a parameterized memory hierarchy, parameterized interconnect buses, and parameterized peripherals. Such SoC platforms must be general enough to be used across several different applications, in order to be economically viable, leading to recent attention to parameterized SoC platforms. Since different applications often have very different power and performance requirements, these parameterized SoC platforms must be optimally configured to meet varied power and performance requirements of a large class of applications. Platform customization is carried out by means of parameters tuning. The embedded system designer has to choose the right value of each system parameter in order to optimize some performance metrics for a given application. Unfortunately, the parameters tuning process is an instance of a combinatorial problem which is known to be NP-hard (Garey and Johnson 1979). It is therefore of strategic importance to define Design Space Exploration (DSE) strategies to search for system configurations that will optimize the desired performance indexes (performance, power consumption, cost, etc.) with a good tradeoff between accuracy and efficiency. There are three main difficulties in DSE: (i) The size of the design space grows as the product of the cardinalities of the variation sets for each parameter; (ii) Evaluation of a single configuration almost always requires the use of simulators or analytical models which are often highly complex; (iii) The objectives being optimized are often conflicting. The result of the exploration will therefore not be a single solution but a set of tradeoffs which make up the Pareto set.

In this paper we present a methodology to address all the three aforementioned problems. The methodology uses an heuristic based on evolutionary computing techniques to explore the design space and fuzzy systems to speedup the evaluation process of any single visited configuration. A multi-objective analysis to tackle with the multi-dimensionality of the design space is also presented. The proposed methodology is applied to the exploration of the design space of a parameterized SoC platform based on a VLIW processor. The use of such platforms for the development of advanced applications, above all in the mobile multimedia area, is a representative testbed to evaluate the methodology. The high degree of parametrization that these platforms feature, combined with the heterogeneous nature of the parameters being investigated, both hardware (architectural, micro-architectural and technology-dependent parameters) and software (compilation strategies and application parameters), demonstrates the scalability of the approach. Experiment results for a multimedia benchmark suite show an improvement in exploration time of over 40% without any appreciable degradation in accuracy.

ACCELERATING DESIGN SPACE EXPLORATION

In (Asca et al. 2004; 2005) has been shown how the use of Genetic Algorithms (GAs) to tackle the problem of DSE gives optimal solutions in terms of both accuracy and efficiency as compared with the state of the art in exploration algorithms. Unfortunately, GA exploration may still be ex-
pensive when a single simulation requires a long compilation and/or execution time. Table 1 shows the total time (compilation + execution) needed for one simulation of some multimedia benchmarks on a Pentium IV Xeon 2.8 GHz Linux Workstation. By a little multiplication we can notice that a few thousands of simulations (just a drop in the immense ocean of feasible configurations) could last from a day to weeks!

<table>
<thead>
<tr>
<th>Benchmark</th>
<th>Description</th>
<th>Sim. time (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>fir</td>
<td>Fir filter</td>
<td>9.1</td>
</tr>
<tr>
<td>ieee810</td>
<td>IEEE-1180 inverse DCT</td>
<td>37.5</td>
</tr>
<tr>
<td>adpcm-enc</td>
<td>Adaptive DPCM speech encoding</td>
<td>22.6</td>
</tr>
<tr>
<td>adpcm-dec</td>
<td>Adaptive DPCM speech decoding</td>
<td>20.2</td>
</tr>
<tr>
<td>mpeg2-dec</td>
<td>MPEG-2 video decoding</td>
<td>113.7</td>
</tr>
<tr>
<td>g721-enc</td>
<td>CCITT G.721 voice compressions</td>
<td>25.9</td>
</tr>
</tbody>
</table>

The primary goal of this work was to define a new approach which could run as few simulations as possible without affecting the very good performance of the GA approach. For this reason we started to develop an intelligent GA approach which has the ability to avoid the simulation of configurations which are estimated to be no good enough to belong the Pareto set. This feature was implemented using a Fuzzy System (FS) to approximate the unknown function from configuration space to objective space. The approach could be described as follows: the GA evolves normally; in the meanwhile the FS learns from simulations until it becomes expert and reliable. From this moment on the GA stops launching simulations and uses the FS to estimate the objectives. Only if the estimated objective values are good enough to enter the Pareto-set will the associated configuration be simulated. The flow chart of the approach is shown in Fig. 1.

We chose a Fuzzy System as an approximator above all because it has cheap additional computational time requirements for the learning process, which are negligible as compared with simulation time. More complex methods, like Neural Networks, need an expensive learning process which could heavily affect time savings without reasonable performance improvements.

In this work we considered true that Fuzzy System is Reliable for estimating results when a certain number of examples were used in the learning process. This way was chosen to test the effectiveness of fuzzy simulation with a low number of examples

**Multi-Objective Genetic Algorithm**

Most real-world problems involve several objectives to be optimized simultaneously, but a single, perfect solution seldom exists for a multi-objective problem. Due to the conflicting nature of at least some of the objectives, only compromise solutions may exist, where improvement in some objectives must always be traded-off against degradation in other objectives. Such solutions are called Pareto-optimal solutions, and there may be many of them for any given problem. For this work we chose SPEA2 (Zitzler et al. 2001), which is very effective in sampling from along the entire Pareto-optimal front and distributing the solutions generated over the trade-off surface.

The chromosome of the GA will then be defined with as many genes as there are free parameters and each gene will be coded according to the set of values it can take (see Figure 2). For each objective to be optimized it is necessary to define the respective measurement functions. These functions, which we will call *objective functions*, frequently represent cost functions to be minimized (e.g. area, power, delay, etc.). Crossover (recombination) and mutation operators produce the offspring. In our specific case, the mutation operator randomly modifies the value of a parameter chosen at random. The crossover between two configuration exchanges the value of two parameters chosen at random. Application of these operators may generate non-valid configurations (i.e. ones that cannot be mapped on the system). A stop criterion...
based on a convergence measure is used to establish when the GA has reached convergence (Ascia et al. 2004).

**Fuzzy Function Approximation**

In our approach we used the well-known (Wang and Mendel 1992) method, which consists of five steps: Step 1 divides the input and output space of the given numerical data into fuzzy regions; Step 2 generates fuzzy rules from the given data; Step 3 assigns a degree to each of the generated rules for the purpose of resolving conflicts among them; Step 4 creates a combined fuzzy rule base on both the generated rules and linguistic rules provided by human experts; Step 5 determines a mapping from the input space to the output space based on the combined fuzzy rule base using a defuzzifying procedure. From Step 1 to 5 it is evident that this method is simple and straightforward, in the sense that it is a one-pass buildup procedure that does not require time-consuming training.

In our implementation the output space could not be divided in Step 1, because we had no information about boundaries. For this reason we used fuzzy rules, which have as consequents a real number $s_j$ associated with all the $M$ outputs:

if $x_1$ is $S_1$ and . . . and $x_N$ is $S_N$ then $y_1 = s_1, \ldots, y_m = s_M$

where $S_i$ are the fuzzy sets associated with the $N$ inputs, which in our implementation are described by Gaussian functions which intersect at a fuzzy degree of 0.5. The choice of the Gaussian function was due to the better performance the fuzzy system gave in our preliminary tests as compared to that with classical triangular sets.

**EXPERIMENTS AND RESULTS**

In this section we present the reference parameterized SOC platform we used to test our methodology, and a set of quality measures used to assess the performance of the proposed approach. As a case study we report the results of the design space exploration for a set of typical multimedia applications (Lee et al. 1997) for optimize both performance and power dissipation.

**Simulation Framework**

To evaluate and compare the performance indexes of different architectures for a specific application, one needs to simulate the architecture running the code of the application. In addition, to make architectural exploration possible both the compiler and the simulator have to be retargetable. EPIC-Explorer (Ascia et al. 2003) is a framework that allows to evaluate any instance of a parameterized VLIW architecture in terms of area, performance and power. It allows the designer to evaluate any application written in C and compiled for any instance of the platform, for this reason it is an excellent testbed for comparison between different design space exploration algorithms. The tunable parameters of the architecture are related to the registers files, functional units, and the memory sub-system. Each of these parameters can be assigned a value from a finite set of values. A complete assignment of values to all the parameters is a configuration. A complete collection of all possible configurations is the configuration space, (also known as the design space). A configuration of the system generates an instance that is simulated and evaluated for a specific application.

The system is divided into a set of sub-blocks as shown in Figure 3. For each sub-block is defined a set of attributes which form the basic constituents of the performance indexes to be optimize. Each set of attributes is estimated by means of a customized fuzzy system. For sake of clarity, let us consider the first level instruction cache (L1$I$ sub-block) whose attributes are the number of misses and the energy consumption. The only information the fuzzy system needs to estimate such attributes are the processor configuration and the L1 instruction cache configuration in place of the overall system configuration. This allow us to reduce drastically the complexity of the estimation problem with the additional effect to improve estimation accuracy.

**Quality Assessment of Pareto set approximations**

It is difficult to define appropriate quality measures for Pareto set approximations, and as a consequence graphical plots were until recently used to compare the outcomes of MOEAs. Nevertheless, quality measures are necessary in order to compare the outcomes of multi-objective optimizers in a quantitative manner, and several quality measures have been proposed in the literature. A review of these is to be found in (Zitzler et al. 2003). We chose the ones we considered most suitable for our context and applied a few modifications:

1. **Distance**, this index explains how close a Pareto set ($P^*_2$) is to a reference set ($P^*_1$). We define the average and maximum distance index as follows:

$$
\text{distance}_{\text{avg}} = \sum_{x_i \in P^*_1} \min_{y_j \in P^*_2} (d(x_i, y_j))
$$

$$
\text{distance}_{\text{max}} = \max_{x_i \in P^*_1} \left( \min_{y_j \in P^*_2} (d(x_i, y_j)) \right)
$$

where $x_i$ and $y_j$ are vectors whose size is equal to the number of objectives $M$ and $d(\bullet, \bullet)$ is the Euclidean distance. To make the index generic and thus be able to compare values obtained with different benchmarks we
applied normalization as follows:

$$ [x_{i,norm}; y_{j,norm}] = \frac{[x_i; y_j] - \max([P_1 \cup P_2])}{\max([P_1 \cup P_2]) - \min([P_1 \cup P_2])} $$

(1)

The lower the value of this index, the more similar the two Pareto sets are.

2. **Cardinality**, this is an integer and represents the configurations selected for inclusion in the Pareto set. A high cardinality indicates that the designer has several solutions at his disposal, so can generally be considered to be positive. However, this index generally needs to be accompanied by others in order to provide significant information, because quantity is not always accompanied by quality.

3. **Hypervolume**, this is the value of the segment describing the Pareto set. To make the value more easily readable and effective, we again normalized the Pareto set in the hypercube $[0, 1]^M$, following (1). A lower value indicates that there are solutions offering a better trade-off.

4. **Pareto Dominance**, the value this index takes is equal to the ratio between the total number of points in Pareto set $P_1$ and the number of points in Pareto set $P_2$ that are also present in a reference Pareto set $R$. In our comparisons there was no absolute reference set, so we have used in its stead the Pareto set deriving from the union between the set obtained via the GA approach and that obtained with the GA-Fuzzy approach. In this way we were able to make a more direct comparison between the two approaches. In this case a higher value obviously corresponds to a better Pareto set.

### Numerical Results

In this subsection we present a comparison between the performance of the classical GA approach and our new GA-Fuzzy approach. In order to minimize differences in results due to the stochastic nature of the GA, we used the same random seed for both approaches, i.e. the two approaches proceeded with the same configurations until the Fuzzy System became reliable and it was used by the GA-Fuzzy approach to estimate the objective values.

The internal and external population for the genetic algorithm were set as comprising 30 or 50 individuals, using a crossover probability of 0.8 and a mutation probability of 0.1. These values were set following an extended tuning phase. The eighteen input parameters, the parameter space and the number of fuzzy sets associated with them are listed in Table 2. Table 2 also gives the number of fuzzy sets associated with each parameter. This number was obtained by means of a GA in a series of preliminary tests.

Table 3 gives the number of configurations simulated by the two approaches and Table 4 presents a comparison between the GA and GA-Fuzzy Pareto sets obtained after 50 generations. Table 5 shows the performance of the two approaches when they ran the same number of simulations. It could be observed from Tables 3 and 5 and Fig. 4 that our proposal yields a similar result to that of the approach based on GAs alone, with a 30-40% saving on calculation time, which may mean several hours or almost a day depending on the benchmark. The consideration that the Pareto sets obtained are equal, as can be seen graphically in the figures, is numerically justified by the short distance between the two sets and the basic equality between the hypervolumes. Table 5 shows that with the same number of simulations, thanks to the fact that it performs more than twice the number of generations, the GA-Fuzzy approach yields a Pareto set which in many points dominates the set provided by the GA, thus being of more benefit to the designer. This is numerically expressed by the higher Pareto Dominance value and the greater number of points in the Pareto set obtained by the GA-Fuzzy approach, which tell us that this set is qualitatively and quantitatively better. Further confirmation of the improvement in performance is given by the fact that in the GA-Fuzzy approach the configurations representing the Pareto end-points (i.e. the objectives of mono-objective optimization) are often solutions that dominate those provided by the GA approach. Another interesting feature of the approach proposed is that at the end of the genetic evolution we obtain a fuzzy system that can approximate any configuration. The designer can exploit this to conduct a more in-depth exploration. Table 6 gives the estimation errors for the fuzzy system obtained after 100 generations on a random set of configurations other than those used in the learning phase. Table 6 shows the time needed for an estimate in a Pentium IV 2.8GHz workstation: despite the great number of rules it is several orders of magnitude shorter than that of a simulation and the degree of accuracy in estimating the objectives is still high.

### CONCLUSION

In this paper we have presented a new Genetic Fuzzy approach to speed up Design Space Exploration. The speedup
Table 3: GA-Fuzzy simulation savings after 50 generations.

<table>
<thead>
<tr>
<th>Benchmark</th>
<th>Sims threshold</th>
<th>Pop. size</th>
<th>Number of simulations GA</th>
<th>GA-F</th>
<th>Savings %</th>
<th>hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>fir</td>
<td>1000</td>
<td>50</td>
<td>2250</td>
<td>1567</td>
<td>30.36</td>
<td>1.73</td>
</tr>
<tr>
<td>adpcm-enc</td>
<td>500</td>
<td>30</td>
<td>1373</td>
<td>862</td>
<td>37.22</td>
<td>3.21</td>
</tr>
<tr>
<td>adpcm-dec</td>
<td>500</td>
<td>30</td>
<td>1381</td>
<td>965</td>
<td>30.07</td>
<td>2.33</td>
</tr>
<tr>
<td>g721-enc</td>
<td>500</td>
<td>30</td>
<td>1354</td>
<td>807</td>
<td>40.39</td>
<td>3.94</td>
</tr>
<tr>
<td>ieee810</td>
<td>700</td>
<td>50</td>
<td>2254</td>
<td>1452</td>
<td>35.46</td>
<td>8.35</td>
</tr>
<tr>
<td>mpeg2-dec</td>
<td>400</td>
<td>30</td>
<td>1389</td>
<td>750</td>
<td>46.00</td>
<td>20.18</td>
</tr>
</tbody>
</table>

Table 4: A comparison between GA and GA-Fuzzy Pareto sets after 50 generations.

<table>
<thead>
<tr>
<th>Benchmark</th>
<th>Distance average</th>
<th>% max</th>
<th>Hypervolume GA</th>
<th>GA-F</th>
</tr>
</thead>
<tbody>
<tr>
<td>fir</td>
<td>0.55</td>
<td>12.38</td>
<td>51.75</td>
<td>50.07</td>
</tr>
<tr>
<td>adpcm-enc</td>
<td>1.52</td>
<td>4.31</td>
<td>50.61</td>
<td>50.25</td>
</tr>
<tr>
<td>adpcm-dec</td>
<td>0.59</td>
<td>4.89</td>
<td>50.48</td>
<td>50.99</td>
</tr>
<tr>
<td>g721-enc</td>
<td>1.10</td>
<td>14.87</td>
<td>49.06</td>
<td>49.36</td>
</tr>
<tr>
<td>ieee810</td>
<td>0.80</td>
<td>4.79</td>
<td>46.79</td>
<td>47.16</td>
</tr>
<tr>
<td>mpeg2-dec</td>
<td>0.68</td>
<td>12.76</td>
<td>45.19</td>
<td>47.50</td>
</tr>
</tbody>
</table>

Table 5: A comparison between GA and GA-Fuzzy Pareto sets after an equal number of simulations.

<table>
<thead>
<tr>
<th>Benchmark</th>
<th>Sims</th>
<th>Cardinality GA</th>
<th>GA-F</th>
<th>Pareto Dominance GA</th>
<th>GA-F</th>
</tr>
</thead>
<tbody>
<tr>
<td>fir</td>
<td>2250</td>
<td>106</td>
<td>107</td>
<td>0.8491</td>
<td>0.9813</td>
</tr>
<tr>
<td>adpcm-enc</td>
<td>1373</td>
<td>48</td>
<td>52</td>
<td>0.6852</td>
<td>0.8654</td>
</tr>
<tr>
<td>adpcm-dec</td>
<td>1381</td>
<td>42</td>
<td>49</td>
<td>0.7857</td>
<td>0.9592</td>
</tr>
<tr>
<td>g721-enc</td>
<td>1354</td>
<td>118</td>
<td>115</td>
<td>0.6186</td>
<td>0.7478</td>
</tr>
<tr>
<td>ieee810</td>
<td>2254</td>
<td>127</td>
<td>171</td>
<td>0.6378</td>
<td>0.7485</td>
</tr>
<tr>
<td>mpeg2-dec</td>
<td>1389</td>
<td>152</td>
<td>158</td>
<td>0.5592</td>
<td>0.8165</td>
</tr>
</tbody>
</table>

Table 6: Approximation performances of the fuzzy system built by GA-Fuzzy on a random unseen set of 10,000 configurations.

<table>
<thead>
<tr>
<th>Benchmark</th>
<th>fir</th>
<th>adpcm enc</th>
<th>adpcm dec</th>
<th>mpeg2 enc</th>
<th>ieee 810</th>
<th>g721 enc</th>
</tr>
</thead>
<tbody>
<tr>
<td>Simulations</td>
<td>2250</td>
<td>1373</td>
<td>1381</td>
<td>1389</td>
<td>2254</td>
<td>1534</td>
</tr>
<tr>
<td>Rules</td>
<td>2172</td>
<td>1382</td>
<td>1342</td>
<td>1354</td>
<td>2168</td>
<td>1313</td>
</tr>
<tr>
<td>Avg Time (ms)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>- Learn</td>
<td>1.79</td>
<td>1.12</td>
<td>1.13</td>
<td>1.27</td>
<td>1.79</td>
<td>1.03</td>
</tr>
<tr>
<td>- Estimate</td>
<td>3.69</td>
<td>2.31</td>
<td>2.33</td>
<td>2.41</td>
<td>3.68</td>
<td>2.26</td>
</tr>
<tr>
<td>Avg Error (%)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Whole Set</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>- Avg Power</td>
<td>7.27</td>
<td>8.69</td>
<td>7.64</td>
<td>18.19</td>
<td>7.28</td>
<td>8.01</td>
</tr>
<tr>
<td>- Exec Time</td>
<td>6.81</td>
<td>9.00</td>
<td>7.66</td>
<td>21.33</td>
<td>6.66</td>
<td>6.93</td>
</tr>
<tr>
<td>Pareto Set</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>- Avg Power</td>
<td>2.26</td>
<td>2.43</td>
<td>2.21</td>
<td>9.55</td>
<td>3.00</td>
<td>2.77</td>
</tr>
<tr>
<td>- Exec Time</td>
<td>4.32</td>
<td>5.67</td>
<td>5.27</td>
<td>10.33</td>
<td>2.52</td>
<td>4.55</td>
</tr>
</tbody>
</table>

Figure 4: adpcm-dec: a comparison between the Pareto sets obtained by GA and the GA-Fuzzy after 50 generations.

is achieved thanks to the ability of the Genetic Fuzzy System to learn from its experience in order to avoid unnecessary simulations. A comparison with the GA approach performed on various multimedia benchmarks showed that integration with the fuzzy system saves a great amount of time and gives the same results, whereas when the length of time is the same the results are better.

REFERENCES


SIMPil-K: a SIMD Reconfigurable Platform Processor for Real-Time Image Processing

F. S. Fabiano, A. Gentile, Member, IEEE, and F. Sorbello, Member, IEEE
Dipartimento di Ingegneria Informatica – DINFO
Università degli Studi di Palermo
Viale delle Scienze, edificio 6
90128 Palermo – ITALY
{Fabiano, Gentile, Sorbello}@unipa.it

KEYWORDS
Image processing, SIMD Pixel Processor, parallel processor, VHDL, FPGA.

ABSTRACT
Image processing is nowadays one of the most important sectors of research in the computer field, for its innumerable applications – vision in robotics, diagnosis in medicine, compression for online data transfer or storage, real time processing, application about safety, among others. The elevated computational cost of most image processing applications makes the researchers experiment new solutions to optimize more and more factors such as execution speed, necessary resources quantity and systems size. In this paper, the architecture of SIMPil-K (K-bit SIMD Pixel Processor) is introduced. SIMPil-K is a SIMD parallel processor that represents a basic hardware support to image processing operations. SIMPil-K has been implemented in VHDL language on Xilinx Integrated System Environment and simulated on Xilinx ModelSim XE II. The next target is the implementation of a prototype on an FPGA board.

INTRODUCTION
The employment of a SIMD parallel processor (Single Instruction/Multiple Data stream), with designing embedded systems for image processing applications, is very suitable, because of their elevated computational cost, and seen the notable possibilities to introduce a considerable level of parallelism in the execution of most digital images elaborations.

SIMPil (SIMD Pixel Processor) architecture was developed for the first time in Atlanta (USA), near Georgia Institute of Technology, under the guide of Prof. D. S. Wills. PICA (Portable Image Computing Architecture) group researchers chose to design a SIMD architecture. SIMD is one of Flynn’s architectural models taxonomy. This taxonomy characterizes four distinct families for computer architecture:
- SISD (Single Instruction/Single Data stream): it is the model used by mono-processor systems, that executes a single instructions stream on a data-set stored in a single memory;
- SIMD (Single Instruction/Multiple Data stream): this category includes the vectorial and array processors; a single CU (Control Unit) produces a single instructions stream that is simultaneously performed by separate PE (Processing Element) on separate data-sets memorized in local memories;
- MISP (Multiple Instruction/Single Data stream): separate instructions streams, produced from separate CU, are performed by the respective PEs on the same data stream, coming from a shared memory; this solution hasn’t yet found any implementation;
- MIMD (Multiple Instruction/Multiple Data stream): each PE performs the instructions stream coming from the respective CU; distinct data streams are simultaneously elaborated. Data sets may be stored either in a shared memory (SM), or in a distributed memory - local

Figure 1 – Flynn’s taxonomy

![Flynn's Taxonomy Diagram]

Figure 2 – SIMD Architecture

![SIMD Architecture Diagram]
memories, DM - or else in a virtual shared memory (VSM); SMPs (Symmetric MultiProcessor), NUMA systems (NonUniform Memory Access) and the clusters belong to this category.

A digital image processing is traditionally submitted to a single processor performing it through the repeated execution of an operation – or a set of operations – on different data. Data are serially delivered to the processor, until it finishes elaborating the whole information concerning the image. By distributing the information on different processing elements, controlled by a single control unit, it is possible to simultaneously perform the operations required by the elaboration on different portions of the image, rather than in sequence. The pixels matrix of the image to be processed is divided into as many square sub-matrix as the PEs; then, every sub-matrix is assigned to a different Processing Element that processes it. The saving in terms of time, in comparison to the same frequency of operation, is evident. PICA group researchers of Georgia Institute of Technology developed the SIMPil project, implemented a SIMPilSim instruction level simulator for Windows, designed and implemented an 8-bit prototype and a 16-bit one according to the VLSI designing canons, implemented in Verilog HDL language a 16-bit processing element. Taking into consideration these outcomes as the starting point, this paper introduces the design of a new SIMD array of processing elements, SIMPil-K, and its implementation in VHDL language; this architecture shows several substantial differences in comparison to its original version. SIMPil-K is a parametric and scalable architecture that provides a valid support to most image processing applications.

RELATED WORK

Most video applications are composed of large blocks of data parallel computation followed by small but significant portions that are better exploited by instruction level parallelism. For example, an image processing application might first search a large image for a few interesting features, then attempt to recognize the features in an object library. The compute intense initial search portion is best performed on a data parallel array. The memory-intense recognition part is best performed on a general-purpose processor that exploits instruction level parallelism. One approach is to use reconfigurable computing fabrics, which exploits the fact that most of the processing time in compute-intensive applications is spent in relatively small portions of the program code. A general-purpose core can be combined therefore with reconfigurable units aimed at accelerating these specific sections of the tasks. Many research projects pursue this architectural solution, like RapID (Cronquist et al. 1997), MorphoSys (Singh et al. 1998), and more recently, Raw (Waingold et al. 1998) and PACT eXtreme Processing Platform (XPPTM) (Baumgarte et al. 2001).

GEORGIA TECH SIMD PIXEL PROCESSOR

The Georgia Tech SIMD Pixel Processor (SIMPil) architecture consists of a mesh of SIMD processors on top of which an array of image sensors is integrated (Gentile et al., 2004a)(Gentile et al., 2004b). A diagram for a 16-bit implementation is illustrated in Fig. 3. Each processing element includes a RISC load/store datapath plus an interface to a 4×4 sensor subarray. A 16-bit datapath has been implemented which includes a 32-bit multiply-accumulator unit, a 16 word register file, and 64 words of local memory (the ISA allows for up to 256 words). The SIMD execution model allows the entire image projected on many PEs to be acquired in a single cycle. Large arrays of SIMPil PEs can be simulated using the SIMPilSim Simulator, an instruction level simulator. Early prototyping efforts have proved the feasibility of direct coupling of a simple processing core with a sensor device (Cat et al., 1996). A 16 bit prototype of a SIMPil PE was designed in 0.8 μm CMOS process and fabricated through MOSIS. A 4,096 PE target system has been used in the simulations. This system is capable to deliver a peak throughput of about 1.5 Tops/sec in a monolithic device, enabling image and video processing applications that are currently unapproachable using today’s portable DSP technology.

![SIMD Pixel Processor Array (SIMPil)](image)

**Figure 3 – The SIMPil architecture**

The SIMPil architecture is designed for image and video processing applications. In general, this class of applications is very computational intensive and requires high throughput to handle the massive data flow in real-time. However, these applications are also characterized by a large degree of data parallelism, which is maximally exploited by focal plane processing. Image frames are available simultaneously at each PE in the system, while retaining their spatial correlation. Image streams can be therefore processed at frame rate, with only nominal amount of memory required at each PE (Gentile et al., 2004a).

**Application suite**

The performance and efficiency of the SIMPil have been tested on a large application suite that spans the target workload. For the SIMPil processing element, an application suite is selected from the DARPA Image Understanding suite (Weens et al., 1991). These applications, listed in Table 1, are expressed in SIMPil assembly language, and executed using an instruction level simulator, SIMPilSim. This
simulator provides execution statistics including dynamic instruction frequency, operand size profiles, PE utilization, and PE memory usage. All applications are executed on a simulated 4,096 processing element system with 16 pixels mapped to each PE for an aggregate 256 x 256 image size. All applications run well within real-time frame-rates and exhibit large system utilization figures (90% or more for most application). Details can be found in (Gentile et al., 2004a).

<table>
<thead>
<tr>
<th>Table I: SIMPil Application Suite</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Image Transform</strong></td>
</tr>
<tr>
<td>Discrete Fourier Transform</td>
</tr>
<tr>
<td>Discrete Cosine Transform</td>
</tr>
<tr>
<td>Discrete Wavelet Transform</td>
</tr>
<tr>
<td>Image Rotation</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Image/Video Compression</th>
<th>Image Analysis</th>
</tr>
</thead>
<tbody>
<tr>
<td>Quantization</td>
<td>Morphological Filtering</td>
</tr>
<tr>
<td>Vector Quantization</td>
<td>Region Representation</td>
</tr>
<tr>
<td>Entropy Coding</td>
<td>Region AutoFocus</td>
</tr>
<tr>
<td>JPEG Compression</td>
<td>K-means Classification</td>
</tr>
<tr>
<td>Motion Estimation</td>
<td></td>
</tr>
<tr>
<td>MPEG Compression</td>
<td></td>
</tr>
</tbody>
</table>

**SIMPIL-K ARCHITECTURE**

To bring SIMPil performance onto embedded platforms, a reconfigurable platform based on FPGA devices is being developed. This platform uses a parameterized SIMPil core (SIMPil-K) described in the VHDL hardware description language. The SIMPil-K platform is an array of Processing Elements (PE) and interconnection registers which can be configured to fit any FPGA device at hand. Figure 4 shows the high-level functional schema of a 4-by-4 SIMPil-K array and its NEWS interconnection network.

Each NEWS register supports communication among a particular node (i.e. PE) and its north and west neighbours. By replicating this model, a NEWS (North, East, West, South) network is obtained, with every node connected to its four neighbours. Moreover, because of its torus configuration, the NEWS network allows nodes laid on opposite edges to communicate too. SIMPil-K receives an instructions stream through a dedicated input port. The instruction stream is then broadcast to each PE.

To upload and download image data, SIMPil-K uses a boundary I/O mechanism, supported by its boundary nodes (i.e. PEs laid on its East/West edge): every east-edge node uploads a K-bit data word from its boundary-input port to the general purpose register file; every west-edge node downloads a K-bit data word from its register file to the boundary-output port. An upload/download operation (one word per node) takes only one clock cycle. Both boundary input and output operations are enabled by a single instruction, XFERTB. When a NEWS transfer instruction arrives, it needs only one clock cycle to transfer the data word from each node to a neighbour one, in a specific direction. When torus-like communication is required - XFERT instruction - N further clock cycles are needed too, with N the size of the N-by-N SIMPil-K array.

**Figure 4 – K-bit 4-by-4 SIMPil-K array**

**Architectural Parameter**

The SIMPil-K platform can be reconfigured by varying a number of architectural parameters, as detailed in Table II. This allows for experimentation with a large set of different system configurations, which is instrumental to determine the appropriate system characteristics for each application environment. AW and RAW parameters set the address space of register file and memory, respectively. PPE specifies the number of image pixels mapped to each PE. The VIW parameter toggle between a fixed instruction width (24 bit) and a variable one (8+K bits). Torus parameter (T) enables/disables NEWS network torus configuration.

<table>
<thead>
<tr>
<th>Table II: SIMPil-K Architectural Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Parameter</strong></td>
</tr>
<tr>
<td>K</td>
</tr>
<tr>
<td>X</td>
</tr>
<tr>
<td>Y</td>
</tr>
<tr>
<td>AW</td>
</tr>
<tr>
<td>RAW</td>
</tr>
<tr>
<td>PPE</td>
</tr>
<tr>
<td>T</td>
</tr>
<tr>
<td>VIW</td>
</tr>
</tbody>
</table>
PROCESSING ELEMENT ARCHITECTURE

The interface of a processing element is depicted in figure 5, below. There are two input ports for clock signals, a reset input port and the instruction stream port. NEWS transfers are carried through the three bidirectional dedicated ports (NEWS ports) which drive three NEWS buses, namely the North/West Bus, East Bus and South Bus. Boundary data input and output are carried through the two dedicated boundary ports.

SIMPiL-Node

Figure 5 – The Processing Element Black Box

The processing element parameterized architecture is described in figure 6.

![Figure 6 – Processing Element K-bit Datapath](image)

There are four communication buses shared by the functional units. All functional units can be reconfigured based on the datapath width selected. A single PE can perform integer operations on K-bits. Dedicated barrel shift unit and multiply-accumulate unit are instrumental to speed-up most image processing kernels. The Sleep Unit verifies and updates the node activity state, thus allowing execution flow control based on each PE local data.

PE’s Architectural Parameters

The SIMPiL-K Processing Element is strongly characterized by some parameters that control its inside architecture and behaviour.

Some of these parameters are the same applied to the whole system.

Three new parameters specify the node position inside the SIMPiL-K array: Boundary Communication Enable (BCE), Boundary Mode (BM) and Node Position (NP). BCE parameter enables or disables the boundary communication function of the node it controls: that is, when BCE is ‘on’, the node lies on the east/west edge of the array.

<p>| Table III: Processing Element Architectural Parameters |
|-----------------|-----------------|-----------------|-----------------|</p>
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Function</th>
<th>Values</th>
<th>Constraints</th>
<th>Def.</th>
</tr>
</thead>
<tbody>
<tr>
<td>K</td>
<td>Word Width</td>
<td>{8,16,32,64}</td>
<td>-</td>
<td>16</td>
</tr>
<tr>
<td>AW</td>
<td>Register File Address Width</td>
<td>AW ∈ N ∩ [1, 16]</td>
<td>if (VIW=off): AW ≤ 4, else: AW ≤ (K/4);</td>
<td>4</td>
</tr>
<tr>
<td>RAW</td>
<td>Local RAM Address Width</td>
<td>RAW ∈ N ∩ [1, 64]</td>
<td>RAW ≤ K;</td>
<td>4</td>
</tr>
<tr>
<td>PPE</td>
<td>Pixel per Processing Element ratio</td>
<td>PPE ∈ N</td>
<td>PPE = p, p ∈ N, PPE ≤ 2^N</td>
<td>8</td>
</tr>
<tr>
<td>T</td>
<td>Torus Enable</td>
<td>T ∈ [off, on]</td>
<td>-</td>
<td>off</td>
</tr>
<tr>
<td>NP</td>
<td>Node Position identification parameter</td>
<td>NP ∈ {n, e, w, s, nw, sw, nw, int}</td>
<td>-</td>
<td>int</td>
</tr>
<tr>
<td>VIW</td>
<td>Variable Instruction Width</td>
<td>VIW ∈ [off, on]</td>
<td>-</td>
<td>off</td>
</tr>
<tr>
<td>BCE</td>
<td>Boundary Communicatio n Enable</td>
<td>BS ∈ [off, on]</td>
<td>-</td>
<td>off</td>
</tr>
<tr>
<td>BM</td>
<td>Boundary Mode Switch</td>
<td>BM ∈ [out, in]</td>
<td>-</td>
<td>off</td>
</tr>
</tbody>
</table>

BM parameter specifies the boundary edge on which the node lies. If BM is ‘out’, the node lies on the west edge, otherwise the node lies on the east one.

NP parameter individuates the node position into the array.

Instructions Format

Figure 7 shows the 24-bit instructions format, when Influence parameter is ‘off’.

![Figure 7 – Standard Instruction Format (I = off)](image)

Every instruction may contain:

- an 8-bit opcode field, a 4-bit destination register field, a 4-bit first source register field and a 4-bit second source register field;
- an 8-bit opcode field, a 4-bit destination register field, a 4-bit first source register field and an 8-bit small immediate field;
- an 8-bit opcode field and a 16-bit large immediate field.
DEST, SRC1 and SRC2 fields support addressing mode to register and indirect addressing mode through register. IMMEDIATE fields support immediate addressing mode. When VIW parameter is 'on', there are an 8-bit opcode field, an AW-bit destination register field, an AW-bit first source register field, an AW-bit second register field, a (K/2)-bit small immediate field and a K-bit large immediate field – that always overlaps other addressing fields.

**PROCESSING ELEMENT FUNCTIONAL UNITS**

**ALU**

ALU block performs fixed-point arithmetical and logical operations: add, subtraction, bitwise AND, bitwise OR and bitwise XOR.

Figure 8 shows the K-bit ALU functional schema.

![Figure 8 – K-bit ALU functional schema](image)

A K-bit BCLA Full Adder executes both arithmetical and logical operations: logical operations are derived by the Generation-Propagation Functions Generate Block inside the adder.

A simple decoder decodes a function code and drives a K-bit logic complementation unit, a K-bit 4-to-1 multiplexer and the input carry toward the full adder; logic complementation unit and input carry drive are needful to perform subtraction.

The K-bit 4-to-1 multiplexer selects the correct result among the adder outputs. It is driven on the C bus.

**Multiplier-Accumulator (MACC)**

The MACC receives two input operands: multiplier and multiplicand.

Each multiplication needs only one clock cycle to be executed. The MACC module takes advantage from the multiplication distributive property: the product between multiplicand and multiplier is equal to the sum of the partial products between the multiplicand and the contained powers of two in the multiplier.

The multiplier is made positive thanks to a 2s complement unit, then all of its (K-1) less significant bits are used as enable for as many left barrel shift units. Each of these units performs a ‘N’ position left shift, where N is the position of the enable bit in the multiplier; this shift is performed only if the respective enable bit is equal to ‘1’; otherwise shift units return zero.

From this shift units, (K-1) partial products are delivered to the first level of a full adders overturned pyramid: it adds all these partial products, and its apex (its last adder) returns an unsigned product.

The 2K-bit unsigned product is complemented, if necessary, and is delivered to the 2K-bit accumulator register: the K less significant bits of that product is also delivered to a 4-to-1 multiplexer too. This multiplexer also receives both K less and most significant bits stored in the accumulator register and selects the correct result.

![Figure 9 – Multiplier-Accumulator Functional Schema](image)

The 2K-bit product stored in the accumulator is complemented and used as feedback for the first adder of the overturned pyramid.

**Shift & Rotate Barrel Unit**

This unit performs left/right arithmetical/logical barrel shift and left/right barrel rotation.

There are two important functional blocks: a left barrel shift unit and a left barrel rotate unit. These units design is the same as the one implemented by the PICA group researchers of Georgia Institute of Technology (Cat et al., 1996). The “shift&rotate” magnitude is made positive thanks to a 2s complement module. This magnitude specifies the number of position of which the operand must be shifted or rotated.

When a right shift, or rotation, is needed, the operand is overturned (i.e. its first and last bits are swapped, its second and next to last bits are swapped too, and so on). After being shifted or rotated, the result are overturned again. When a left shift, or rotation, is needed, all the overturning units are disabled.
Local Memory

The PE’s Local Memory is a single-ported $2^{KN} \times K$-bit distributed RAM. It has got a RAW-bit address port, a K-bit write port and a K-bit read one. It also receives a clock enable signal, a clock signal and a function code which selects the correct operation (load 0/1 store). Write is triggered by the falling edge of the clock signal. Read is asynchronous.

When a load operation occurs, this unit sends the addressed word to the register file through the C bus. When a store operation occurs, this unit reads a data word from the B bus and stores it in the addressed memory cell.

Register File

The PE’s Register File is a dual-ported $2^{AW} \times K$-bit distributed RAM. It has got a K-bit write port – connected to the C bus – and two K-bit read ones – connected to the A and B buses. It also receives a write enable signal, a clock signal, an AW-bit write address and two AW-bit read ones. Write is triggered by the falling edge of the clock, and read is asynchronous.

Sleep Unit

The Sleep Unit controls, verifies and updates the Processing Element activity status. Moreover, it sends an activity status bit to the decoder, to enable or disable the instructions execution.

This unit stores a particular register, called Sleep Vector: it is a K-bit mask that gives information about the sleep status of the node and makes possible to wake up the processing element.

The sleep status is updated every time a sleep or wake-up instruction occurs; the Sleep Vector is updated too. When a conditional sleep instruction occurs, the Sleep Unit verifies the instruction predicate and, if that is true, puts the node to sleep. When a conditional wake-up instruction occurs, if the bitwise AND between Sleep Vector and a K-bit mask – red from B bus – is not zero, this unit wakes up the node and clears or updates the Sleep Vector.

PROCESSING ELEMENT EMPLOYED RESOURCES ON FPGA

The SIMPil-K Processing Element has been simulated and synthesized on FPGA; synthesis statistics about employed resources has been generated and analyzed.

Figure 12 shows resources use percentage achieved by implementing several K-bit Processing Element versions on a one million gates FPGA: particularly, 8-bit, 16-bit, 32-bit and 64-bit Processing Element are analyzed.

By increasing the system word width (K) the Processing Element resources employment on FPGA exponentially increases, especially because of ALU and MACC logic growing complexity.

SIMPIL-K EMPLOYED RESOURCES ON FPGA

SIMPil-K system has been simulated and synthesized on
FPGA; synthesis statistics about employed resources has been generated and analyzed.

![Graph showing processing element used resources on one million gates FPGA](image)

**Figure 12 – Processing Element used resources on one million gates FPGA**

Figure 13 shows resources use percentage achieved by implementing several 16-bit SIMPl-K versions on an eight million gates FPGA; particularly, 2-by-2, 4-by-4 and 8-by-8 16-bit SIMPl-K array are analyzed. As the graph illustrates, by increasing the array sizes there is an exponential increase in resources employment on FPGA, due to exponential growth of the number of processing elements and NEWS registers.

![Graph showing SIMPl-K used resources on eight million gates FPGA](image)

**Figure 13 – SIMPl-K used resources on eight million gates FPGA**

**CONCLUSION**

In this paper the SIMPl-K (K-bit SIMD Pixel Processor) architecture is detailed.

SIMPl-K is a valid support to new and future implementations and extensions.

SIMPl-K NEWS interconnection network supports a torus-mode communication (i.e. communication between nodes laid on opposite edges).

In a future implementation the optimization effort will be addressed to a 64-bit architecture.

Moreover, SIMPl-K needs an Array Control Unit (ACU, i.e. a unit that generates the instructions stream and monitors the activity of the SIMD array), which still has to be implemented.

**REFERENCES**


Bhasker, J. *A VHDL Primer*. Prentice Hall PTR. 1999;


Hwang, K. *Computer Arithmetic: principles, architecture, and design*. John Wiley & Sons, Inc. 1979;


Stallings, W. *Architettura e Organizzazione dei Calcolatori*. Jackson Libri. 2002;


SCALABLE CIRCUIT MODELS FOR PASSIVE HIGH-SPEED INTERCONNECTION STRUCTURES

Tom Dhaene
University of Antwerp
Middelheimlaan 1
2020 Antwerp, Belgium
E-mail: Tom.Dhaene@ua.ac.be

ABSTRACT

An automated circuit-modeling tool is presented for arbitrary planar transmission lines. The tool builds compact, parameterized, analytical models based on multiple full-wave 2D electro-magnetic (EM) simulations. The transmission line parameters are stored as a multidimensional function of frequency and geometrical parameters. The modeling algorithm combines adaptive data selecting and modeling techniques. The circuit models combine EM-accuracy and generality, and circuit simulation speed and flexibility.

1. INTRODUCTION

Multiconductor transmission line structures form a basic building block of microwave and RF integrated circuits. The modeling of their behavior at microwave and millimeter wave frequencies is essential to the correct functioning of these devices (Dhaene and De Zutter 1992; Dhaene et al. 1994). Their presence is certainly not restricted to MIC’s (Microwave Integrated Circuits) and MMIC’s (Monolithic MIC’s). Due to the increasing bitrates in digital broadband systems and in computers, the behavior of multiconductor busses on boards and backplanes becomes increasingly important.

Accurate parameterized circuit models for arbitrary transmission line structures are required for the design and optimization of high-speed electronic circuits. Several numerical 2D EM techniques (such as the method of moments (Momentum 1995-2006)) can be used to accurately model transmission lines. However, most numerical EM techniques require a significant amount of expertise and computer resources, so that they are often only used for verification purposes. On the other hand circuit simulators are very fast, and offer a lot of different analysis possibilities. However, the number of available analytical transmission line models is limited, and the accuracy is not always guaranteed up to RF or microwave frequencies.

We developed a new automated tool for building parameterized circuit models of general passive transmission line structures with user-defined accuracy. The analytical models represent the transmission line parameters as a multidimensional function of frequency and geometrical parameters.

The models are based on full-wave 2D EM simulations, and can easily be incorporated in circuit simulators. This brings EM-accuracy and generality in the circuit simulator, without sacrificing speed. The model generation process is fully automated. Data points are selected efficiently and model complexity is automatically adapted. The algorithm consists of an adaptive modeling loop (section 2) and an adaptive sample selection loop (section 3). An example is given to illustrate the technique (section 4).

2. ADAPTIVE MODEL BUILDING ALGORITHM

Manuscripts Coupled transmission lines are 2D structures, and they are fully characterized by their length \( l \), and by their impedance matrix \( Z_{\alpha\beta} = R_{\alpha\beta} + j \omega L_{\alpha\beta} \) and the admittance matrix \( Y_{\alpha\beta} = G_{\alpha\beta} + j \omega C_{\alpha\beta} \) per unit length (Dhaene and De Zutter 1992)-(Dhaene et al. 1994). The circuit parameters are generated using the 2D solver of the commercially available full-wave electro-magnetic simulator ADS Momentum (Momentum 1995-2006).

The transmission line parameters \( R, L, G \) and \( C \) are approximated by a weighted sum of multidimensional orthonormal polynomials (multinomials) \( P_m \). The multinomials only depend on the coordinate \( \bar{x} \) in the multidimensional parameter space \( R \), while the weights \( C_m \) only depend on the frequency \( f \).

\[
RLGC(\bar{f}, \bar{x}) = A(\bar{f}, \bar{x}) = \sum_{m=1}^{M} C_m(\bar{f}) P_m(\bar{x})
\]  

(1)

The weights \( C_m \) are calculated by fitting equation (1) on a set of \( D \) data points \( \{\bar{x}_d, S(\bar{f}, \bar{x}_d)\} \) (with \( d = 1, \ldots, D \)). The number of multinomials \( M \) is adaptively increased until the error function \( E(\bar{f}, \bar{x}) = ||RLGC(\bar{f}, \bar{x}) - A(\bar{f}, \bar{x})|| \) is lower than a user-defined accuracy level in all the data points. For numerical stability and efficiency reasons orthonormal multinomials are used (De Geest et al. 1999; Dhaene et al. 2001).

3. ADAPTIVE DATA SELECTION ALGORITHM

The modeling process starts with an initial set of data points in the multidimensional parameter space. New data points are added adaptively until the user-defined accuracy level is guaranteed.

The process of selecting data points and building models in an adaptive way is called reflective exploration (Beyer and Smieja 1996). Reflective exploration is useful when the process that provides the data is very costly, which is the case for full-wave EM simulators. Reflective exploration requires reflective functions that are used to select a new data point.
The difference between 2 consecutive approximate models (with different order \( M \) in (1)) is used as a reflective function. A new data point is selected near the maximum of the reflective function. No new data points are added if the magnitude of the reflective function is smaller than the user-defined accuracy level (over the whole parameter space).

Physical rules are also checked. If the approximate modeling function \( A(f,\bar{x}) \) violates certain physical rules, a new data point is chosen where the criteria are violated the most.

Furthermore, at least one data point is chosen in the close vicinity of local minima and maxima of the modeling function \( A(f,\bar{x}) \) over the parameter space of interest.

The complete flowchart of the adaptive modeling algorithm is given in Fig. 1.

![Figure 1. Adaptive modeling and sampling flowchart](image)

Figure 1. Adaptive modeling and sampling flowchart

![Figure 2. Single transmission line: cross section.](image)

Figure 2. Single transmission line: cross section.

### Table 1. Single transmission line: parameter ranges

<table>
<thead>
<tr>
<th>variable</th>
<th>min</th>
<th>max</th>
</tr>
</thead>
<tbody>
<tr>
<td>Width</td>
<td>20 ( \mu \text{m} )</td>
<td>100 ( \mu \text{m} )</td>
</tr>
<tr>
<td>( f )</td>
<td>0 ( \text{GHz} )</td>
<td>60 ( \text{GHz} )</td>
</tr>
</tbody>
</table>

4. EXAMPLE: MICROSTRIP TRANSMISSION LINES

The automated modeling tool was used to generate analytical circuit models for a single transmission line, and 2 coupled transmission lines. A GaAs microstrip substrate was used (\( h = 100 \mu \text{m}, \varepsilon_r = 12.9 \)).

The parameter ranges of the single transmission line (Fig. 2) circuit are shown in Table 1. The new adaptive modeling tool selected 5 data points (= discrete 2D layouts) over the parameter range in an adaptive way, and grouped all \( RLGC \)-parameter data all in one global, compact, analytical model. In Fig. 3, the reflection coefficient \( S_{11} \) (reference impedance 50\( \Omega \)) is shown as a function of Width and frequency for a line of 1 mm long. Note the wave behavior along the frequency axis, and the (almost) zero reflection if Width = 73\( \mu \text{m} \) (corresponding with a characteristic impedance of 50\( \Omega \)).

![Figure 3. Single transmission line (1 mm): \( S_{11}(W,f) \).](image)

**Figure 3. Single transmission line (1 mm): \( S_{11}(W,f) \).**

![Figure 4. Coupled transmission lines: cross section.](image)

Figure 4. Coupled transmission lines: cross section.

### Table 2. Coupled transmission lines: parameter ranges

<table>
<thead>
<tr>
<th>variable</th>
<th>min</th>
<th>max</th>
</tr>
</thead>
<tbody>
<tr>
<td>Width</td>
<td>40 ( \mu \text{m} )</td>
<td>100 ( \mu \text{m} )</td>
</tr>
<tr>
<td>Spacing</td>
<td>5 ( \mu \text{m} )</td>
<td>50 ( \mu \text{m} )</td>
</tr>
<tr>
<td>( f )</td>
<td>0 ( \text{GHz} )</td>
<td>60 ( \text{GHz} )</td>
</tr>
</tbody>
</table>
The parameter ranges of the coupled transmission lines (Fig. 4) circuit are shown in Table 2. The automated modeling tool selected 35 data points (corresponding with 35 transmission line configurations) over the parameter range of interest in an adaptive way, and grouped all RLGC-parameter data in one global, compact, analytical matrix model. In Figure 5, the inductance per unit length $L_{11}$, and the capacitance per unit length $C_{11}$ are shown as a function Width, Spacing and frequency. As expected, the capacitance increases if Width and frequency increase, and Spacing decreases.

5. CONCLUSIONS

A new adaptive technique was presented for building parameterized models for general passive planar interconnection structures. The models are based on full-wave EM simulations, and have a user-defined accuracy. Once generated, the analytical models can be grouped in a library, and incorporated in a circuit simulator where they can be used for simulation, design and optimization purposes. Two microstrip examples were given to illustrate the new technique.

REFERENCES


Momentum, Advanced Design System (ADS), Agilent Technologies, Santa Rosa, CA.
SIMULATION OF CIRCADIAN RHYTHM PHASE LAGS BETWEEN SERVERS

Peter Lawrence
Clayton School of Information Technology
Monash University, Clayton Campus
Victoria 3800 AUSTRALIA
Email: peter.lawrence@infotech.monash.edu.au

ABSTRACT

It has been suggested that simulating processes involving human operators with stationary stochastic models is invalid and may be responsible for the failure of some simulation projects. Previous studies by the author have failed to find any significant errors produced by ignoring the daily cyclic efficiency variation of humans but these studies all assumed everyone in the system was tied to the same cycle. Results present here however, suggest that the introduction of phase lags between the cycles of individuals can significantly impact the reliability of simulation models based on stationary human performance statistics.

INTRODUCTION

Discrete Event Simulation (DES) is a well worn tool for the study of many types of transaction processing systems and has proven very successful over the years. However, it has been claimed that, while such simulations are excellent for highly automated systems, they often fail when human activity is involved (Baines and Kay 2002). What simplifies the modelling of automated systems is that timing elements such as server cycle times operate within small tolerances but when processing systems involve human operators complex timing irregularities can occur. In discrete event simulations these irregularities are generally modelled using random numbers sampled from an appropriately chosen probability distribution and any inaccuracy in output is often the result of poorly chosen input distributions.

This traditional approach usually involves reducing the performance characteristics of human operators to a statistical distribution that is stationary. Where possible, the distribution is chosen by gathering field data by timing real operations in situ, aggregating the results and fitting a theoretical (or even empirical) distribution to the histogram. Unless it is obvious, in most cases no attempt is made to identify any possible non-stationary behaviour before the data are aggregated.

Similarly, attempts are rarely made to identify whether timing varies between individual operators and indeed, an attempt to do so would likely be fraught with industrial relations consequences. Of course, in environments where cycle times are not driven by automated elements such as conveyor speed, performance variation between individuals is to be expected. Were it possible to identify individual performance levels this source of error in simulations could be removed by assigning different cycle time distributions to each worker.

Independent of performance variation between individuals there is still the issue of assuming stationary timing behaviour which remains a convenient but questionable practice.

The Psychology literature contains many studies that highlight the non-stationary nature of human performance but there appears to be no accurate quantitative model of human efficiency on the horizon (Baines et al. 2003a, 2003b). For this reason the author has conducted studies of the impact that plausible variations in human operator timing might have on a balanced flow line (Lawrence 2004) and in a job shop (Lawrence 2005). Those studies indicated that the traditional approach of aggregated, time-invariant distributions produces only minor errors in such systems. However, in those studies it was always assumed that any time-variance in performance was shared by all workers and that they all “ran to the same clock”. This is not necessarily a valid assumption. The work reported here attempts to investigate the impact of cyclic timing variations between human operators.

TIME-VARIANT HUMAN PERFORMANCE

It is well accepted that human performance varies during the course of the working day (Smith 1989, 1992). Many human biological functions, such as body temperature, follow a natural cycle and it is generally accepted that this impacts behaviour and performance in some manner. Work in this area appears to have begun early in the twentieth century and was aimed at establishing the optimum time of day for academic teaching (Carrier & Monk 2000).

Aschoff & Wever (1962: cited in Scott 1994) established that these biological rhythms have a period of about 25 hours but they are entrained to the 24-hour day by external time cues (zeitgeber) and have thus become known as circadian rhythms (Scott 1994).

Scott cites much evidence to suggest a connection between this circadian rhythm for body temperature and human performance. Importantly, studies have suggested that the phase relationships between biological and performance cycles are not necessarily fixed. For example, tasks with a small cognitive load seem to have a performance variation that tracks the circadian body temperature cycle (Kleitman 1963) while high memory load tasks are phase-shifted from...
this cycle. Folkard et al. (1976) suggested high memory tasks to be in anti-phase with the temperature rhythm and Monk’s (1982) analysis used Folkard’s data to postulate a medium to high memory load performance cycle that peaks 3-hours ahead of the low load performance cycle.

Monk also noted that superposed on this circadian effect is the additional factor of number of hours worked. Smith (1989, 1992) also discusses the impact of meals on performance including the so called “post-lunch dip”. It is also possible for individual workers to have variations between their rhythms. Shift work can alter circadian cycles and workers on rotating shifts can have cycles that substantially flatten out (Harrington 2001).

It is therefore quite plausible that through the nature of their tasks or their recent sleep patterns adjacent workers in a processing environment could exhibit cyclic performance variations that are out of phase with one another.

Of course, this phase relationship would remain unnoticed and un-modelled in a normal simulation project. It was the aim of this study to see if these effects are significant enough to affect the final accuracy of the modelling.

THE MODEL

The system investigated was very simple. It involved two servers, each with a single, separate human operator, sequentially processing incoming jobs. Server 1 had an infinite supply of jobs so this it was always 100% utilised while server 2 processed jobs as they became available from server 1.

As with previous studies, the processing time at each server was assumed to be a log-normal random variate with a standard deviation (σ) of 7% of the mean (μ). Thus,

$$\sigma = 0.07 \mu$$

To model human time-variant performance it was assumed that this mean would vary in accordance with circadian rhythms and so μ was re-calculated at the commencement of each processing event in a simulation run. To allow the investigation of phase lags between the circadian rhythms of the two workers a phase component was also introduced so that each log-normal random variate was generated as required with a mean for worker i of,

$$\mu_i = 1 + \alpha \cos \left( \frac{2\pi t}{1440} + \phi_i \right)$$

where t is the simulation run time in minutes, $\phi_i$ is the starting phase angle for worker i and $\alpha$ is the amplitude of the circadian variation in the cycle time. The amplitude factor, $\alpha$, was varied between runs but for any given experiment both workers had the same $\alpha$. It is certainly possible for different workers to have different $\alpha$ values but the effect of that variable has not been investigated here.

A typical series of cycle time is shown in Figure 1.

![Figure 1 A typical series of process cycle times for worker 2 with a circadian amplitude of $\alpha=0.2$](image)

A model of this system was built in Arena and all replications were run for 24 hours of simulated operation so a complete circadian cycle was modelled.

EXPERIMENTS

Several series of experiments were conducted to test:

- the effect of circadian amplitude on system performance,
- the effect of phase differences in circadian rhythms between the two workers and
- whether or not the traditional approach of aggregating observations to build a simulation based on time-invariant cycle time distributions would produce accurate results, as was the case in previous studies by the author.

For the purposes of these experiments system performance was defined by the following three metrics:

- total daily production volume (P),
- utilisation of worker 2 (U) and
- the average time a job waited between process 1 and process 2 (Q).

In each case 100 replications were made of each experiment and 99% confidence interval estimates made of each of the three performance measures.

Effect of Circadian Amplitude

The model was run with no phase lag between the workers,

$$\phi_1 = \phi_2 = 0$$

and various circadian amplitudes.

Effect of Phase Lag

To determine the effect of phase lag, experiments were performed where $\phi_1$ was held at zero against various values of $\phi_2$. This was done for circadian amplitudes of $\alpha = 0.05$, 0.1 and 0.15.
Time-Invariant Distributions

To test this approach the model containing the circadian variation was considered to be the “real” system and is henceforth referred to as the \( \Psi \)-model. The \( \Psi \)-model was run for 2 complete, 24-hour replications and the cycle times for each worker over this period were recorded. This mimics the process of visiting the real system and recording cycle times that is commonly carried out in simulation studies.

These data were aggregated in to a single distribution histogram for each worker and the Arena Input Analyser was used to find a best fit analytical distribution.

A second model was then constructed, known as the M-model, that was identical to the \( \Psi \)-model except that all cycle times were generated from the aggregated distributions and so contained no time-variant component. The M-model therefore contained no specific phase lag information, although worker 2’s aggregated cycle time distribution would not necessarily be the same as that of worker 1. These experiments were carried out with a circadian amplitude of \( \alpha = 0.15 \).

RESULTS

The results in Table 1 are from the \( \Psi \)-model and show the effect of \( \alpha \) on system performance.

<table>
<thead>
<tr>
<th>( \alpha )</th>
<th>Production</th>
<th>Utilisation</th>
<th>Waiting(mins)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>P</td>
<td>U</td>
<td>Q</td>
</tr>
<tr>
<td>0.00</td>
<td>1435.3</td>
<td>0.997</td>
<td>0.000</td>
</tr>
<tr>
<td>0.05</td>
<td>1437.1</td>
<td>0.997</td>
<td>0.000</td>
</tr>
<tr>
<td>0.10</td>
<td>1442.7</td>
<td>0.997</td>
<td>0.000</td>
</tr>
<tr>
<td>0.15</td>
<td>1451.8</td>
<td>0.997</td>
<td>0.000</td>
</tr>
<tr>
<td>0.20</td>
<td>1465.0</td>
<td>0.997</td>
<td>0.000</td>
</tr>
</tbody>
</table>

The circadian variation results in a small but statistically significant increase in production as the amplitude increases. This may be ostensibly surprising since each worker has completed an entire circadian cycle during each replication and so the overall mean cycle time should be the same for any given \( \alpha \). In fact this is not the case. Since each new cycle time is calculated when processing begins the cycle times are not sampled at a constant rate. When the cycle time is high the sampling rate drops and then increases again as the cycle time diminishes. This means that more samples are taken in the high efficiency part of the circadian cycle than in the low efficiency period which reduces the overall mean cycle time. The greater the circadian amplitude, the greater the reduction in mean time.

Circadian variation had no apparent effect upon worker 2’s utilisation but this is close to unity even without the variation. There was however, a marked increase in the queuing time between the two servers.

The results presented in Table 2 are also for the \( \Psi \)-model and show the effect of phase lag between workers on system performance. It is apparent that phase lag has a significant effect on production volume and queuing time and even a small but noticeable impact on the utilisation of the second worker.

It is therefore clear that phase lag can have an impact on the overall behaviour of the system and the results for the M-models in Table 3 assess how well a time-invariant distribution can model these effects.

Comparison between the \( \Psi \) and M-models in Figure 2 shows that the time-invariant distribution does a good job of modelling the increase in production volume as \( \alpha \) increases. It is true that the confidence intervals between the \( \Psi \) and M-models do not overlap and hence we can reject the null hypothesis that these data come from the same distribution. However, the estimates are so close that for all practical purposes they would be seen as an excellent match. Of course, this increase in production has already been identified as a result of the decrease in mean cycle time so it is not surprising that the time-invariant distribution can handle this effect. Looking at the worker utilisation results in Figure 3 shows a similar story. The confidence intervals do not overlap but in most cases an error like this in utilisation estimates, being in the third decimal place, would once again be an excellent result.

The results for queuing time in Figure 4 however, do not at appear to be very good as they show an error well in excess of 10%. Of course, the absolute value of this error, in the order of 3 minutes, represents the time to process 3 units and is therefore consistent with the size of the error in production volume. Interestingly two different approaches to measuring the same error can yield such apparently different results. When looked at as 3 units of production out of 1440 per day it seems small but, under certain circumstances, an extra 3 minute wait may be critical.

When phase lag effects are introduced in the system the M-model approach fails, on a grand scale, to produce acceptable estimates. The comparison for production volume in Figure 5 shows that shifting circadian phase of the second operator can produce drops in production in excess of 50 units (3.5%) while the M-Model fails to clearly indicate any trend at all.

Phase lag also causes a small but measurable drop in utilisation (Figure 6) that the time-invariant distribution fails to predict.

The effect on Queueing time, which once again reflects the discrepancy in the production volume estimate, clearly highlights the failure of the model. Expecting queuing delays of the order of 5 minutes but actually experiencing 45 minute delays could be disastrous in many circumstances, such as when the entities being processed are perishable goods or customers.
Table 2 (a) Daily production volumes, (b) worker 2 utilisation and (c) inter-process queueing time as a function of phase lag between workers for various circadian amplitudes

<table>
<thead>
<tr>
<th>( \phi_2 (\pi) )</th>
<th>( \alpha = 0.00 )</th>
<th>( \alpha = 0.05 )</th>
<th>( \alpha = 0.10 )</th>
<th>( \alpha = 0.15 )</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>( P ) 99%±</td>
<td>( P ) 99%±</td>
<td>( P ) 99%±</td>
<td>( P ) 99%±</td>
</tr>
<tr>
<td>0.00</td>
<td>1435.3 0.60</td>
<td>1437.1 0.59</td>
<td>1442.7 0.58</td>
<td>1451.8 0.60</td>
</tr>
<tr>
<td>0.25</td>
<td>1435.3 0.60</td>
<td>1422.6 0.73</td>
<td>1410.8 0.68</td>
<td>1401.9 0.69</td>
</tr>
<tr>
<td>0.50</td>
<td>1435.3 0.60</td>
<td>1411.7 0.67</td>
<td>1388.2 0.61</td>
<td>1367.4 0.72</td>
</tr>
<tr>
<td>0.75</td>
<td>1435.3 0.60</td>
<td>1409.7 0.67</td>
<td>1384.6 0.78</td>
<td>1362.0 0.72</td>
</tr>
<tr>
<td>1.00</td>
<td>1435.3 0.60</td>
<td>1416.6 0.64</td>
<td>1399.2 0.70</td>
<td>1384.9 0.58</td>
</tr>
<tr>
<td>1.25</td>
<td>1435.3 0.60</td>
<td>1426.6 0.67</td>
<td>1420.3 0.61</td>
<td>1417.6 0.70</td>
</tr>
<tr>
<td>1.50</td>
<td>1435.3 0.60</td>
<td>1435.0 0.59</td>
<td>1436.6 0.69</td>
<td>1442.3 0.75</td>
</tr>
<tr>
<td>1.75</td>
<td>1435.3 0.60</td>
<td>1437.9 0.58</td>
<td>1443.2 0.59</td>
<td>1452.1 0.63</td>
</tr>
<tr>
<td>2.00</td>
<td>1435.3 0.60</td>
<td>1437.1 0.59</td>
<td>1442.7 0.58</td>
<td>1451.8 0.60</td>
</tr>
</tbody>
</table>

(b)

<table>
<thead>
<tr>
<th>( \phi_2 (\pi) )</th>
<th>( \alpha = 0.00 )</th>
<th>( \alpha = 0.05 )</th>
<th>( \alpha = 0.10 )</th>
<th>( \alpha = 0.15 )</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>( U ) 99%±</td>
<td>( U ) 99%±</td>
<td>( U ) 99%±</td>
<td>( U ) 99%±</td>
</tr>
<tr>
<td>0.00</td>
<td>0.997 0.000</td>
<td>0.997 0.000</td>
<td>0.997 0.000</td>
<td>0.997 0.000</td>
</tr>
<tr>
<td>0.25</td>
<td>0.997 0.000</td>
<td>0.987 0.000</td>
<td>0.976 0.000</td>
<td>0.965 0.000</td>
</tr>
<tr>
<td>0.50</td>
<td>0.997 0.000</td>
<td>0.980 0.000</td>
<td>0.962 0.000</td>
<td>0.945 0.000</td>
</tr>
<tr>
<td>0.75</td>
<td>0.997 0.000</td>
<td>0.979 0.000</td>
<td>0.961 0.000</td>
<td>0.943 0.000</td>
</tr>
<tr>
<td>1.00</td>
<td>0.997 0.000</td>
<td>0.984 0.000</td>
<td>0.970 0.000</td>
<td>0.956 0.000</td>
</tr>
<tr>
<td>1.25</td>
<td>0.997 0.000</td>
<td>0.990 0.000</td>
<td>0.982 0.000</td>
<td>0.975 0.000</td>
</tr>
<tr>
<td>1.50</td>
<td>0.997 0.000</td>
<td>0.996 0.000</td>
<td>0.993 0.000</td>
<td>0.990 0.000</td>
</tr>
<tr>
<td>1.75</td>
<td>0.997 0.000</td>
<td>0.998 0.000</td>
<td>0.997 0.000</td>
<td>0.997 0.000</td>
</tr>
<tr>
<td>2.00</td>
<td>0.997 0.000</td>
<td>0.997 0.000</td>
<td>0.997 0.000</td>
<td>0.997 0.000</td>
</tr>
</tbody>
</table>

(c)

<table>
<thead>
<tr>
<th>( \phi_2 (\pi) )</th>
<th>( \alpha = 0.00 )</th>
<th>( \alpha = 0.05 )</th>
<th>( \alpha = 0.10 )</th>
<th>( \alpha = 0.15 )</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>( Q ) (mins.)</td>
<td>( Q ) (mins.)</td>
<td>( Q ) (mins.)</td>
<td>( Q ) (mins.)</td>
</tr>
<tr>
<td>0.00</td>
<td>1.84 0.28</td>
<td>1.82 0.28</td>
<td>1.77 0.28</td>
<td>1.82 0.28</td>
</tr>
<tr>
<td>0.25</td>
<td>1.84 0.28</td>
<td>6.02 0.24</td>
<td>11.57 0.24</td>
<td>17.25 0.26</td>
</tr>
<tr>
<td>0.50</td>
<td>1.84 0.28</td>
<td>12.42 0.25</td>
<td>24.68 0.27</td>
<td>36.99 0.26</td>
</tr>
<tr>
<td>0.75</td>
<td>1.84 0.28</td>
<td>18.42 0.32</td>
<td>36.54 0.32</td>
<td>54.69 0.30</td>
</tr>
<tr>
<td>1.00</td>
<td>1.84 0.28</td>
<td>21.55 0.32</td>
<td>42.68 0.36</td>
<td>64.10 0.32</td>
</tr>
<tr>
<td>1.25</td>
<td>1.84 0.28</td>
<td>21.00 0.38</td>
<td>40.72 0.34</td>
<td>61.27 0.42</td>
</tr>
<tr>
<td>1.50</td>
<td>1.84 0.28</td>
<td>16.54 0.41</td>
<td>31.84 0.45</td>
<td>46.80 0.46</td>
</tr>
<tr>
<td>1.75</td>
<td>1.84 0.28</td>
<td>9.45 0.47</td>
<td>17.54 0.45</td>
<td>25.34 0.51</td>
</tr>
<tr>
<td>2.00</td>
<td>1.84 0.28</td>
<td>1.82 0.28</td>
<td>1.77 0.28</td>
<td>1.82 0.28</td>
</tr>
</tbody>
</table>

Table 3 results from the M-model with aggregated distributions from the Ψ-model with \( \alpha=0.1 \) and various phase lags from zero to \( 2\pi \).
**CONCLUSIONS**

Though previous studies have failed to show significant deficiencies in the use of time-invariant distributions to model human performance those studies all assumed that all operators were running on the same clock. There are however, many plausible scenarios where different operators may be at different points in their circadian efficiency cycles for the task at hand. The results presented here show that this can invalidate the traditional discrete event simulation methodology of using time-invariant statistical distributions to model human irregularity.
REFERENCES


305
SEMICONDUCTOR MANUFACTURING
SIMULATION-BASED ASSESSMENT OF LOT PLAN REPAIR STRATEGIES IN SEMICONDUCTOR MANUFACTURING

Ilka Habenicht
Technical University Ilmenau
Institute of Information Systems
Helmholtzplatz 3
98693 Ilmenau, Germany
E-mail: ilka.habenicht@tu-ilmenau.de

KEYWORDS
Semiconductor manufacturing, lot plan repair, beam search.

ABSTRACT
In this paper we describe lot plan repair strategies for semiconductor manufacturing using a finite-capacity beam search algorithm. We introduce two methods of lot repair. We describe a simulation environment which is used to evaluate the performance of these strategies. We show some results from computational experiments with the simulation test-bed described.

INTRODUCTION
Electronic industry has become the largest industry in the world. The manufacturing of integrated circuits (IC) on silicon wafers is the most important area in this industry and a very complex production process (cf. Atherton and Atherton 1995, Uzsoy et al. 1992). The production of a circuit of medium complexity requires between 250-500 process steps on 50-120 different types of equipment. A mix of different process types, i.e., batch processes or single wafer production, are typical for this production. The production of ICs takes place in clean room environments and requires a very expensive equipment.

A single product moves through the production process in lots. Each lot consists of several wafers. It is possible to place up to 800 circuits on a single wafer. The circuits have a layered nature. A decomposition of the entire wafer fabrication into work areas takes place. The recursive structure of the production process is one of the main reasons for difficulties in planning, scheduling, and controlling wafer fabrication.

Particularly, the execution of lot plans and schedules during the production process is frequently interrupted by different stochastic and uncertain events, i.e., postprocessing steps, unplanned maintenance and repair of equipment, or stochastic influences of production and setup times. Therefore, a planning and scheduling system must be able to consider such uncertainties (Toba 2000). A complete regeneration of lot plans leads to unfavourable dispatching decisions. A given setup or batching decision can become unfavourable when the lot plan completely changes. Therefore we evaluate two approaches for partial lot plan repair in this paper.

The paper is organized as follows. In the next section, we describe the lot planning problem for semiconductor manufacturing and the basics of the beam search algorithm used for initial planning. We present two approaches for lot plan repair. The last part of the paper deals with the description of the simulation-based evaluation of these approaches. We describe the simulation test-bed and the design of experiments. Finally, we present some results of simulation experiments.

PROBLEM DESCRIPTION AND MODELLING

Problem description
In this section, we will formulate the lot planning problem addressed in this paper. First, an aggregation of process steps into operations takes place. An operation is defined as set of consecutive process steps of one product. Usually, an operation consists of three or four process steps. More formally, if a product \( P_k \) has a process flow \( s_k := (s_{k1}, \ldots, s_{kn}) \), then we replace \( s_k \) by the sequence \( o_k := (o_{k1}, \ldots, o_{km}) \) of operations. These operations are considered instead of single process steps. The decision, which process steps form an operation, is influenced by the decomposition of the shop floor into different work areas.

The lot planning problem can be formulated as follows. Given a set of lots \( \{L_1, \ldots, L_I\} \). We assume that each lot \( L_i \), \( i = 1, \ldots, I \), has to perform the operations \( o_i := (o_{i1}, \ldots, o_{im}) \).

By \( o_{i1} \), we denote the first operation of lot \( L_i \), which has to be performed and by \( o_{im} \) the last operation. We denote the due date of lot \( L_i \) by \( d_i \). We are interested in determining the start date and the end date of each operation of the lot.

There are a number of references dealing with lot plan repair. Vieira et al. give an overview over different replanning strategies, policies and methods (Vieira et al. 2002). In replanning, two aspects are important. The first aspect is choosing a replanning policy or answering the question “when to replan?” We can differ between periodic, event-driven and hybrid policies. Most of references for planning in semiconductor manufacturing deal with a periodic policy, i.e., (Liao et al. 1996). Lot plans are rebuilt after a defined time period. In (Toba 2000) an event-driven lot planning approach is described. This policy includes a replanning of lots controlled by defined events. In a dynamic environment like semiconductor manufacturing, only a view events which lead to a unfeasible lot plans should be considered. Ayteg et al. conclude that hybrid policies are achieved in practice which includes a periodic and an event-driven policy (Ayteg et al. 2005).
The second aspect of replanning is the replanning method. Vieira et al. differ between right-shift replanning, partial replanning and complete generation. In (Habenicht and Mönch 2002) a periodic replanning with a complete plan regeneration is considered. Problem of complete regeneration is the amount of time for replanning and the plans of all lots can completely differ. This aspect can lead to unfavourable setup and batching decisions. In this paper, we choose a partial replanning method. Only those operations are replanned where the lot plan became unfeasible.

Initial Planning

We use the beam search algorithm described in (Habenicht and Mönch 2002) for initial planning. The planning horizon is divided into time buckets. The length of these time buckets increases to the end of the planning horizon. The capacity restrictions for planning an operation can be formulated for each time interval.

The essence of the beam search algorithm is the search tree shown in Figure 1. Every node of the tree represents a time bucket $T_m$. The search tree is partitioned into levels. Each level $k$ represents planning one operation of the lot. All nodes of this level have distance $(k-1)$ to the root. Beam width $\beta_k$ defines the maximum number of nodes belonging to level $k$.

![Figure 1: Search tree of the beam search algorithm](image)

The goal of the algorithm is to determine start and end dates for the operations of each single lot. So the algorithm is formed by two loops. The outer loop ranks the lots by their importance. The importance of lot $L_i$ with due date $d_i$ at the point of time $t$ where the lot planning is performed can be calculated as follows

$$slack_i = d_i - t - \sum_{k=n_i}^m p_{ki}$$  

The inner loop of the algorithm uses the beam search algorithm for lot planning.

RESCHEDULING METHODS

In (Habenicht and Mönch 2002) a rolling lot planning approach is described, where lot plans were built from scratch with every lot planning step. In this section, we introduce two approaches for partial lot plan repair with the beam search algorithm. A shift of the planning horizon and the connected time buckets has to take place. We illustrate how the lot plans are adapted to the new time buckets. We describe how to determine those lots whose lot plans are unfeasible.

Lot Plan Repair

Method I

The first method considers only those lots whose plans has to be repaired. In this case, we use the described beam search algorithm to plan those lots with taking into account the actual capacity restrictions on the shop floor. Figure 2 shows the outer loop of the approach. All lots with unfeasible plans are sorted by their importance (calculated by the slack index of equation (1)). The capacity restrictions are formulated taking into account those lots with a feasible lot plan. The inner loop contains the beam search algorithm described in (Habenicht and Mönch 2002).

![Figure 2: Planning lots if possible (Method I)](image)

The importance of the lots with unfeasible lot plans is not considered. Therefore, the start and end dates of operations of lots with a very small slack index can be delayed far into the future. This causes an increasing tardiness of those lots.

Method II

The second method allows a lot plan repair by deleting and rebuilding plans of lots with lower importance. The algorithm has to be adapted in two ways. Firstly, the beam search algorithm includes the possibility of deleting lot plans or part of lot plans of lots with a larger slack index:

We assume, operation $o_{ij}$ was assigned at level $(k-1)$ of the search tree to the time bucket $T_m$.

- Then we search for a possible time bucket for planning the following operation $o_{ij+1}$. Therefore, we branch into nodes on the level $k$ with possible time buckets for scheduling $o_{ij+1}$. We try to plan $o_{ij+1}$ without violating the capacity restrictions given by the tool groups required for processing $o_{ij+1}$.
• If the capacity restrictions of a time bucket at level \( k \) are not satisfied we delete the lot plans of lots with lower importance and try to plan without violating the updated capacity restrictions given by the tool groups required for processing \( o_{g+1} \).

• We update the capacity restrictions of the tool group of operation \( o_{g+1} \) if planning into a time bucket at level \( k \) is possible.

• A backtracking step has to take place if a planning step in a time bucket at level \( k \) is not possible due to the violation of capacity constraints. That means, the operation \( o_{g} \) has to be removed from \( T_{m} \) and to be planned to one of the following time buckets of \( T_{m} \). The capacity restrictions of the tool group of operation \( o_{g} \) are updated. The backtracking scheme allows more than one consecutive backtracking step.

The adapted beam search algorithm has consequences for forming the outer loop of this method, shown in Figure 3. The lots, whose plans are deleted by planning another lot with higher importance, must be included to the lot list by considering the slack value of these lots.

![Diagram](image)

**Figure 3: Planning lots as early as possible (Method II)**

### Feasibility of Lot Plans

For lot planning and lot plan repair, we consider a planning horizon divided into time buckets with an increasing length. If a lot planning step takes place after a defined period of time \( \Delta t \), the time scheme of the planning horizon has to be shifted to \( \Delta t \) into the future.

For lot plan repair, we have to shift the time scheme just as well. That means, we have to adapt all lot plans to the new time scheme in this way that the calculated start and end dates are still valid. The capacity restrictions have to be reformulated for the new time buckets.

There is one problem connected with the shift of the time scheme. Because of the increasing lengths of the time buckets, it is possible that an operation of a lot, which is planned into a time bucket of larger length, now is assigned to a time bucket of smaller length. This can lead to unsatisfied capacity restrictions for this operation. In this case, the lot has to be replanned, too.

The first step of the described lot plan repair methods is the determination of those lots whose lot plans are unfeasible. We assume two criteria for identifying unfeasible lot plans:

- a lot is related to the given start and end dates,
- capacity restrictions are not satisfied.

The former criteria concerns only one lot. The latter criteria affects all lots whose operations are assigned to the interval and the tool group where the capacity restriction is not satisfied. In this case, we replan as many lots as we need for satisfying the capacity restrictions. We begin with the lot with the lowest importance given by formula (1).

### PERFORMANCE ASSESSMENT

#### Simulation Model

We use a discrete-event simulation tool (AutoSched AP 8.1) and a simulation model of a small wafer fab to evaluate the performance of the described lot repair methods. The architecture of the simulation test bed is described in (Mönch et al. 2002).

The used simulation model is adapted from a small complexity model suggested by the researchers from INTEL (El Adl et al. 1997). The process flow of the original model is organized in two layers. It contains three tool groups and three product routes including six steps. One tool group is a batch processing one. Our model contains three work areas. Each of these work areas contains the tool groups of the original model. The process flow is now organized in six layers, where the processing of two layers is situated in each work area. Therefore the model contains 18 process steps on nine tool groups. We use a slack based dispatching rule. The slack of a process step is calculated with the help of the given start and end dates of the lot plan. The model is initialized by using a work in process (WIP) distribution of the wafer fab. The length of a simulation run was 200 days in our experiments.

#### Design of Experiments and Performance Measurement

Based on previous experiences with lot planning, we expect an influence of the work load of the manufacturing system, the due date settings, and the distribution of the lot weights on the performance measures.

We set the due dates \( d_{i} \) for lot \( L_{i} \) with release date \( r_{i} \) through the relation

\[
  d_{i} = r_{i} + \sum_{k=1}^{m} p_{ki} .
\]

We denote by \( p_{ki} \) the processing times of the process steps and by \( f \) the flow factor. The distributions \( D_{1} \) and \( D_{2} \) are used to assign the lot weights:

\[
  D_{1} = \begin{cases} 
    w_{1} = 1 \text{ with } p_{1} = 0.5 \\
    w_{2} = 5 \text{ with } p_{2} = 0.35 \text{ and } D_{2} = \begin{cases} 
    w_{1} = 2 \text{ with } p_{2} = 0.45 \\
    w_{2} = 10 \text{ with } p_{3} = 0.15
  \end{cases} \\
  w_{1} = 10 \text{ with } p_{3} = 0.05
\]
We chose the system parameters given in Table 1 for the system evaluation. We use the duple (flow factor, priority distribution) in order to indicate the used factor combination. We use a moderate work load of the fab.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>I</th>
<th>II</th>
</tr>
</thead>
<tbody>
<tr>
<td>Flow Factor</td>
<td>1.2</td>
<td>1.5</td>
</tr>
<tr>
<td>Weight Distribution</td>
<td>$D_1$</td>
<td>$D_2$</td>
</tr>
</tbody>
</table>

We consider two different method parameters. The first parameter is the time period $\Delta t$ for lot planning and lot repair. The second parameter is the tolerance $\theta$ for the tardiness of lots to the lot schedule. A lot plan of lot $L_i$ with the current operation $o_{in}$ is feasible at time $t$ if the following constraint is satisfied

$$t + p_{in} - d_{in} \leq \theta \cdot p_{in}.$$  \hspace{1cm} (3)

We denote by $p_{in}$ the whole processing time of the current operation, by $p_{in}$ the left processing time of the current operation, and by $d_{in}$ the end date of the current operation given by the lot plan.

The following performance measures were used:

- The total weighted tardiness of the lots released and finished within the planning horizon under consideration. We denote this quantity by $TWT$.
- Average predictability of the plans. Because the planned completion times of the lots will change over time, we use the completion time calculated by the first plan that contains the lot $L_i$ as a fixed reference, denoted by $C^P_i$. We define the positive deviation of the (first) plan as follows:

$$D_1^+ = \max \left\{ C^P_i - C^P_i - d^l_{in}, 0 \right\},$$  \hspace{1cm} (4)

where $C^P_i$ is defined as the real completion time of lot $L_i$. We sum over the quantity (4) for all corresponding lots to derive the performance measure for predictability. The corresponding quantity is denoted by $D^*$. 

- An important performance measure for evaluating lot plans is the stability $S$ of the lot plans. This performance measure gives information how much two successive lot plans differs from each other. We define the stability of a lot plan containing $I$ lots.

$$S := \frac{1}{I} \sum_{i=1}^{I} \left| d^\text{act}_{i} - d^\text{last}_{i} \right|.$$  \hspace{1cm} (5)

We denote by $d^\text{act}_{i}$ the due date of the lot $L_i$ given by the actual lot plan and by $d^\text{last}_{i}$ the due date given by the last lot plan.

- The average number of lot which are replanned by one planning step is denoted by $\#\text{lots}$. The number of lots which are replanned gives information about the amount of time needed for planning.

### Computational Results

We perform a number of experiments in order to evaluate how partial replanning influences the performance measures in comparison to complete lot plan regeneration. Table 2 gives an overview of the results of the experiments with building the lot plans from scratch. The resulting performance measures are presented in terms of the ratio of the performance measure value of the heuristic and the experiment by replanning all lots after a time period of four hours rule performance measure value.

<table>
<thead>
<tr>
<th>Parameters</th>
<th>$\Delta t$</th>
<th>$TWT$</th>
<th>$D^+$</th>
<th>$S$</th>
<th>$#\text{lots}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>I-I</td>
<td>4h</td>
<td>1.0000</td>
<td>1.0000</td>
<td>1.0000</td>
<td>1.0000</td>
</tr>
<tr>
<td></td>
<td>8h</td>
<td>1.0210</td>
<td>0.8559</td>
<td>0.7768</td>
<td>1.0469</td>
</tr>
<tr>
<td></td>
<td>24h</td>
<td>1.0840</td>
<td>1.4696</td>
<td>0.4646</td>
<td>1.2031</td>
</tr>
<tr>
<td>II-I</td>
<td>4h</td>
<td>1.0000</td>
<td>1.0000</td>
<td>1.0000</td>
<td>1.0000</td>
</tr>
<tr>
<td></td>
<td>8h</td>
<td>1.0122</td>
<td>1.0092</td>
<td>0.7570</td>
<td>1.0469</td>
</tr>
<tr>
<td></td>
<td>24h</td>
<td>1.0467</td>
<td>4.0750</td>
<td>0.4267</td>
<td>1.2031</td>
</tr>
<tr>
<td>II-II</td>
<td>4h</td>
<td>1.0000</td>
<td>1.0000</td>
<td>1.0000</td>
<td>1.0000</td>
</tr>
<tr>
<td></td>
<td>8h</td>
<td>1.0164</td>
<td>0.9792</td>
<td>0.8034</td>
<td>1.0308</td>
</tr>
<tr>
<td></td>
<td>24h</td>
<td>1.1733</td>
<td>0.6932</td>
<td>0.5140</td>
<td>1.1846</td>
</tr>
<tr>
<td></td>
<td>4h</td>
<td>1.0000</td>
<td>1.0000</td>
<td>1.0000</td>
<td>1.0000</td>
</tr>
<tr>
<td></td>
<td>4h</td>
<td>1.0519</td>
<td>1.3915</td>
<td>0.7834</td>
<td>1.0469</td>
</tr>
<tr>
<td></td>
<td>24h</td>
<td>1.1117</td>
<td>1.5790</td>
<td>0.4734</td>
<td>1.1875</td>
</tr>
</tbody>
</table>

It turns out from this table that the performance measures depend on the time period for planning. The increasing amount of lots results from the larger number of forecast lots which are planned. We plan all forecast lots which will enter the fab during the next planning period.

In the second series of experiments we used Method I for lot plan repair. We present the corresponding results in Table 3. We see that the method cannot outperform a complete lot plan regeneration considering the total weighted tardiness, especially for the parameter setting I-I. If a lot with a small slack index is replanned, its operations are delayed far into the future. Therefore urgent lots are belated. The predictability and the stability of the lot plan becomes better. This is caused by the small number of lots which are replanned.

<table>
<thead>
<tr>
<th>Parameters</th>
<th>$\Delta t$</th>
<th>$\theta$</th>
<th>$TWT$</th>
<th>$D^+$</th>
<th>$S$</th>
<th>$#\text{lots}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>I-I</td>
<td>4h</td>
<td>0.00</td>
<td>1.4038</td>
<td>0.3123</td>
<td>0.4435</td>
<td>0.2813</td>
</tr>
<tr>
<td></td>
<td>8h</td>
<td>0.00</td>
<td>1.4364</td>
<td>0.3141</td>
<td>0.3441</td>
<td>0.4219</td>
</tr>
<tr>
<td></td>
<td>24h</td>
<td>0.00</td>
<td>1.4020</td>
<td>0.4052</td>
<td>0.1799</td>
<td>0.5156</td>
</tr>
<tr>
<td>II-I</td>
<td>4h</td>
<td>0.00</td>
<td>1.1367</td>
<td>0.6971</td>
<td>0.4283</td>
<td>0.2500</td>
</tr>
<tr>
<td></td>
<td>8h</td>
<td>0.00</td>
<td>1.1556</td>
<td>0.9604</td>
<td>0.4257</td>
<td>0.4375</td>
</tr>
<tr>
<td></td>
<td>24h</td>
<td>0.00</td>
<td>1.0824</td>
<td>0.9505</td>
<td>0.1425</td>
<td>0.4063</td>
</tr>
<tr>
<td>II-II</td>
<td>4h</td>
<td>0.00</td>
<td>1.0466</td>
<td>0.5091</td>
<td>0.4348</td>
<td>0.2769</td>
</tr>
<tr>
<td></td>
<td>8h</td>
<td>0.00</td>
<td>1.1961</td>
<td>0.5188</td>
<td>0.3252</td>
<td>0.4000</td>
</tr>
<tr>
<td></td>
<td>24h</td>
<td>0.00</td>
<td>1.2177</td>
<td>0.2890</td>
<td>0.1889</td>
<td>0.4923</td>
</tr>
</tbody>
</table>

We represent the results of the series of experiments using Method II for lot plan repair in Table 4. This method...
outperforms Method I but not a complete lot regeneration. The predictability of the lot plans becomes worse. This is caused by the often replanning of lots with low importance. These lots are processed later than their first plan predicts.

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Δt</th>
<th>Δθ</th>
<th>TWT</th>
<th>D⁺</th>
<th>S</th>
<th># lots</th>
</tr>
</thead>
<tbody>
<tr>
<td>I-I</td>
<td>4h</td>
<td>0.00</td>
<td>1.3075</td>
<td>0.2343</td>
<td>0.8981</td>
<td>0.7031</td>
</tr>
<tr>
<td></td>
<td>8h</td>
<td>0.00</td>
<td>1.3036</td>
<td>0.2750</td>
<td>0.6055</td>
<td>0.7656</td>
</tr>
<tr>
<td></td>
<td>24h</td>
<td>0.00</td>
<td>1.4309</td>
<td>0.9018</td>
<td>0.2635</td>
<td>0.8593</td>
</tr>
<tr>
<td>I-II</td>
<td>4h</td>
<td>0.00</td>
<td>1.0751</td>
<td>1.3812</td>
<td>0.9038</td>
<td>0.6563</td>
</tr>
<tr>
<td></td>
<td>8h</td>
<td>0.00</td>
<td>1.0839</td>
<td>1.4283</td>
<td>0.6208</td>
<td>0.7188</td>
</tr>
<tr>
<td></td>
<td>24h</td>
<td>0.00</td>
<td>1.1111</td>
<td>1.4572</td>
<td>0.2399</td>
<td>0.7188</td>
</tr>
<tr>
<td>II-I</td>
<td>4h</td>
<td>0.00</td>
<td>1.0641</td>
<td>1.3646</td>
<td>0.8239</td>
<td>0.6769</td>
</tr>
<tr>
<td></td>
<td>8h</td>
<td>0.00</td>
<td>1.0470</td>
<td>1.3186</td>
<td>0.5662</td>
<td>0.7385</td>
</tr>
<tr>
<td></td>
<td>24h</td>
<td>0.00</td>
<td>1.1443</td>
<td>0.9491</td>
<td>0.3092</td>
<td>0.8615</td>
</tr>
<tr>
<td>II-II</td>
<td>4h</td>
<td>0.00</td>
<td>1.0707</td>
<td>2.1937</td>
<td>0.8878</td>
<td>0.6563</td>
</tr>
<tr>
<td></td>
<td>8h</td>
<td>0.00</td>
<td>1.0699</td>
<td>2.4085</td>
<td>0.6015</td>
<td>0.7188</td>
</tr>
<tr>
<td></td>
<td>24h</td>
<td>0.00</td>
<td>1.0518</td>
<td>0.9237</td>
<td>0.2545</td>
<td>0.7188</td>
</tr>
</tbody>
</table>

Figure 4 shows the dependence of the performance measures on the tolerance for the feasibility constraint (3) for factor combination I-I and a planning period of 4 hours. The total weighted tardiness increases with increasing tolerance. This effect is caused by the decreasing amount of lots which are replanned.

**CONCLUSIONS AND FUTURE RESEARCH**

In this paper, we described two approaches for lot plan repair based on a finite-capacity beam-search algorithm. The first method replans only those lots whose plans are unfeasible. The second method considers the importance of the lots and allows to replan lots with low importance. We describe the simulation test-bed used for performance assessment of the described methods. We present some results of simulation experiments.

The methods do not perform better than a complete lot plan regeneration considering the total weighted tardiness. The stability of the lot plans is better using partial lot plan repair. We expect better results for complexer fabs than the considered model. We will evaluate both methods of lot plan repair with models of higher complexity.

The present work can be extended to hybrid replanning policies. We expect an improvement of the performance by replanning on events which cause an unfeasible lot plan, i.e. machine breakdowns.

**REFERENCES**


**BIOGRAPHY**

ILKA HABENICHT is a Ph.D. student in the Department of Information Systems at the Technical University of Ilmenau, Germany. She received a master’s degree in business related engineering from the Technical University of Ilmenau, Germany. Her research interests are in production control of semiconductor wafer fabrication facilities and simulation.
PERFORMANCE EVALUATION OF A SHIFTING BOTTLENECK HEURISTIC FOR A MULTI-PRODUCT WAFERFAB
BY DISCRETE-EVENT SIMULATION

Jens Zimmermann, Lars Mönch
Chair of Enterprise-wide Software Systems
Department of Mathematics and Computer Science
FernUniversität in Hagen
Universitätstraße 1, D-58097 Hagen, Germany
E-mail: [jens.zimmermannlars.moench]@fernuni-hagen.de

KEYWORDS
Shifting Bottleneck Heuristic, Semiconductor Manufacturing, Multi-Product, Discrete-Event Simulation, Performance Assessment

ABSTRACT
In this paper, we present the results of a simulation study for the performance evaluation of a shifting bottleneck heuristic in a multi-product semiconductor wafer fabrication facility (waferfab). The shifting bottleneck decomposes the overall scheduling problem into scheduling problems for single tool groups. The solutions of these smaller scheduling problems are connected via a disjunctive graph. The shifting bottleneck heuristic is applied in a rolling horizon manner. In this simulation study, we extend previous work on the evaluation of the shifting bottleneck heuristic from the two-product case to the case of a larger number of products.

INTRODUCTION
In semiconductor manufacturing, integrated circuits are produced on silicon wafers. This type of manufacturing is very capital intensive. Lots are the moving entities in a waferfab. Each lot contains a fixed number of wafers. The process conditions are very complex (Urzoy et al. 1994, Atherton and Atherton 1995, Schöning and Fowler 2000). We have to deal with parallel machines, different types of processes like batch processes and single wafer processes, sequence dependent setup times, prescribed customer due dates for the lots, and reentrant process flows. Batch machines can process several lots at the same time on the same machine. Very often, we also have to face with an over time changing product mix including a large number of different products. Production control strategies based on dispatching rules are very common in semiconductor manufacturing. A dispatching rules selects a lot among the lots waiting in front of a tool group. Dispatching rules are conceptually easy to understand and can be implemented with less effort on the shop-floor. However, usually they are not able to adapt to different circumstances on the shop-floor. They are not able to take different criteria into account.

In contrast, scheduling approaches consider several lots and machines at the same time. A scheduling approach determines a schedule, i.e., an assignment of lots to tools and a sequencing of these lots on each machine. Usually, scheduling approaches are able to deal with multiple objectives. They are able to adapt quite easily to different systems conditions.

In this paper, we consider a shifting bottleneck heuristic applied to a model of a multi-product waferfab. We are interested in determining the influence of the number of different products on the solution quality of the shifting bottleneck heuristic. We are mainly interested in minimizing total weighted tardiness of the lots, i.e., in increasing on-time delivery performance. However, we also have to consider throughput and cycle time because these two measures are also important from a practical point of view.

The paper is organized as follows. In the next section, we summarize the shifting bottleneck heuristic that is used in this study. Then, we describe the simulation model and our experimental design. We present and discuss the results from simulation experiments in the last section of the paper.

SHIFTING BOTTLENECK HEURISTIC

In this section, we summarize main principles of the shifting bottleneck heuristic. The shifting bottleneck heuristic is a prominent decomposition approach for job shops. It decomposes the overall scheduling problem into scheduling problems for single tool groups. A scheduling graph connects the results of the scheduling problems for single tool groups and provides a view on the overall problem. The main steps of the original shifting bottleneck heuristic can be described as follows (Mason et al. 2002):

1. Denote the set of all tool groups by $M$. We use the rotation $M_0$ for the set of tool groups that have already been sequenced or scheduled. Initially, set $M_0 := \emptyset$.

2. Identify and solve the subproblems for each tool group $i \in M - M_0$.
3. Identify a critical tool group \( k \in M - M_0 \).
4. Sequence the critical tool group using the subproblem solution obtained by Step 2 by incorporating the related conjunctive arcs into the scheduling graph. Set \( M_\text{adj} := M_0 \cup \{k\} \) for update purposes.
5. (Optionally) re-optimize the schedule for each tool group \( m \in M_\text{adj} - k \) by exploiting the information provided by the newly added disjunctive arcs for tool group \( k \).
6. If \( M = M_\text{adj} \), terminate the heuristic. Otherwise, go to Step 2.

We use Figure 1 to show a disjunctive graph for the lots 1, 2, 3, and the machines 1, 2, 3, 4. The notation \([i, j]\) is used for a node that is associated with the processing of lot \( j \) on machine \( i \). The processing time of lot \( j \) on machine \( i \) is denoted by \( p_{ij} \). The ready time of lot \( j \) is denoted by \( r_j \). An artificial start node \( s \) and an ending node \( e \) are introduced. Furthermore, we use the node \( v_j \) in order to represent the due dates of lot \( j \). We refer to (Pinedo 2002) for a more detailed description of disjunctive graphs.

![Disjunctive Graph](image)

Figure 1: Disjunctive Graph

Mason et al. (2002) suggest several modifications of the original shifting bottleneck heuristic. The approach deals with batching issues and reentrant flows by including additional arcs into the scheduling graph. Batching and sequence dependent setup issues are taken into account on the subproblem solution level.

The performance measure of interest is the total weighted tardiness. This measure is defined as follows:

\[
TWT := \sum_{j=1}^{n} w_j (c_j - d_j)^+ ,
\]

where we denote by
- \( w_j \) : weight of lot \( j \),
- \( c_j \) : completion time of lot \( j \),
- \( d_j \) : due date of lot \( j \),
- \( n \) : number of completed lots.

We set for abbreviation \( x^+ := \max(0, x) \) throughout the rest of the paper.

So far, only little is known on the performance of the shifting bottleneck heuristic for wafer fabs and especially in multi-products situation. Only the limited case of two products is investigated in (Fowler et al. 2006). Many papers that deal with the shifting bottleneck heuristic consider only static problems (cf. Mason et al. 2002 or the case of scheduling wafer fabs). The multi-product case is interesting because the number of products influences the batching and has a great influence on the entire dynamic of the wafer fab.

**SIMULATION-BASED PERFORMANCE ASSESSMENT**

In this section, we discuss first the used simulation model and the software infrastructure. Then we describe the design of experiments. Finally, we present the results of computational experiments with the shifting bottleneck heuristic in a multi-product environment.

**Simulation Infrastructure**

We use the architecture suggested by Mönch et al. (2002) for carry out the experiments. The center point of this architecture is a blackboard type data layer that is between the simulation engine AutoSched AP 7.3 and the shifting bottleneck heuristic. The data layer acts as a mirror of the manufacturing process. It contains all the relevant information like tool status and lot status to build the scheduling graph for the shifting bottleneck heuristic.

**Used Simulation Models**

In this study, we use a reduced version of the MIMAC I test data set (Fowler and Robinson 1995). It contains the 11 leading bottlenecks of the MIMAC I data set. Therefore, our model is formed by 45 machines. In (Fowler et al. 2006), the number of products is limited by considering only two product flows. In this paper, we extend these investigations by considering more products.

For the case of eight products, we copy product flow \( A \) and \( B \) four times and for sixteen products eight times respectively. We use 16 products of product flow \( A \) and product flow \( B \) respectively in case of 32 products. Each product flow created in this way represents a certain product.

Lots that represent different products cannot be batched together and therefore influence the dynamic of the waferfab to a large extend.

**Design of Experiments**

We expect that the solution quality depends on the due dates and the load of the manufacturing system. Therefore, we choose these two factors in our experiments. Furthermore, we also expect that the number of different products has an impact on the solution quality.

We use two different weight schemes for the lots. The discrete distribution \( D_t \) describes a situation where a
large amount of lots has small or medium weight. \( D_1 \) is
given by:

\[
D_1 := \begin{cases} 
  w_j = 1 & p_j = 0.5, \\
  w_j = 5 & p_j = 0.35, \\
  w_j = 10 & p_j = 0.15.
\end{cases} \tag{2}
\]

The weight distribution \( D_2 \) is used to mimic a situation
where only a very small portion of lots has an extremely
large weight. \( D_2 \) is defined as follows:

\[
D_2 := \begin{cases} 
  w_j = 1 & p_j = 0.5, \\
  w_j = 2 & p_j = 0.45, \\
  w_j = 10 & p_j = 0.05.
\end{cases} \tag{3}
\]

The product mix specifies the number of lots of the basic
two product flows. The 1:1 mix situation means that the
time between the start of the lots of the different
product flows is equal. In the 2:1 mix situation, the time
between the lot starts is two times longer for the second
product flow as for the first product flow. There are two
times more lots of product flow A in this situation in the
waferfab. We proceed in an analogous way in the case of
4, 8, 16, and 32 products by considering the fact that the
products are created by copying the product flows A and B
respectively.

To determine the due dates of the lots, we use the flow
factor to calculate these due dates. We set the due date
of lot \( j \) by

\[
d_j := FF \sum_{k=1}^{w_j} p_{jk} + r_j , \tag{4}
\]

where we denote by \( p_{jk} \) the processing time of processing
step \( k \) of lot \( j \) and \( u_t \) denotes the number of processing
steps of lot \( j \). We use the abbreviation \( FF \) for the
flow factor. The possible amount of waiting time that is
spent by lot \( j \) is therefore given by

\[
\left( FF - 1 \right) \sum_{k=1}^{w_j} p_{jk} . \tag{5}
\]

We compare the results of the shifting bottleneck heuristic
with results that are obtained by using common
dispatching rules. In our experiments, we use the First
In First Out (FIFO) rule, the critical ratio (CR) rule, the
Highest Priority rule (HP), Earliest Due Date (EDD),
and a combination of HP and EDD (HP & EDD).
CR ranks the lots according to the ratio of remaining
time until the due date and the raw processing time. The
HP rules selects lots with greatest weight \( w_j \) first. EDD
selects the lot with the smallest due date \( d_j \) next. HP &
EDD uses first HP and breaks ties by applying EDD.

To reduce the simulation time we start each simulation
with a work in process (WIP). We perform deterministic
simulation runs for 50 days.

The subproblem solution procedure of the SBH is the
Apparent Tardiness Cost rule (ATC) in case of non-
batching machines and the Batched Apparent Tardiness
Cost rule (BATC) in case of batch machines.

We consider high and moderate loaded systems in our
experiments. We summarize the used design of experiments in Table
1.

<table>
<thead>
<tr>
<th>Table 1: Design of Experiments</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Factor</strong></td>
</tr>
<tr>
<td>Weight Setting</td>
</tr>
<tr>
<td>Product Mix</td>
</tr>
<tr>
<td>Due Date Setting</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>Load of the System</td>
</tr>
<tr>
<td>Number of Products</td>
</tr>
<tr>
<td>Dispatching Rule</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>Total</td>
</tr>
</tbody>
</table>

Totally, we have to perform 480 simulation runs for the
full factorial design. Each of them takes between 60 to
270 minutes on a Pentium 4 2 GHz PC with 512 MB
memory. The time required for computing depends on the
number of products. A higher number of products leads to a higher computing time, because of the increasing WIP.

Results of Computational Experiments
In this section, we present the results of our simulation
experiments.

We consider the number of completed lots (removing the
initial WIP lots) (denoted by \#lots), the total
weighted tardiness TWT, and the average weighted tardiness
denoted by AWT. This quantity is defined by the expression

\[
AWT := \frac{1}{n} \sum_{j=1}^{n} w_j \left( c_j - d_j \right) . \tag{6}
\]

In order to overcome the difficulties with absolute
values, we also use relative values denoted by REL. REL is
defined as the ratio of the TWT value obtained by the
SBH or a dispatching rule and the corresponding FIFO
value.

In Table 2, we consider a situation with two products,
the weight distribution \( D_1 \), the product mix 2:1, a flow
factor of 1.4, and a high load. In this situation, SBH
outperforms all dispatching rules. However, the difference
between SBH and the best dispatching rule (HP &
EDD) is rather small. The number of completed lots is basically the same. Table 3 shows the corresponding results of the experiments in the same situation but with 32 products instead of only two products.

Table 2: Results for the Two Product Case

<table>
<thead>
<tr>
<th>Rule</th>
<th>#lots</th>
<th>TWT (in h)</th>
<th>AWT (in h)</th>
<th>REL</th>
</tr>
</thead>
<tbody>
<tr>
<td>FIFO</td>
<td>551</td>
<td>66288.20</td>
<td>120.3053</td>
<td>1.0000</td>
</tr>
<tr>
<td>CR</td>
<td>558</td>
<td>30441.82</td>
<td>54.5552</td>
<td>0.4535</td>
</tr>
<tr>
<td>HP</td>
<td>549</td>
<td>22895.53</td>
<td>41.7041</td>
<td>0.3467</td>
</tr>
<tr>
<td>EDD</td>
<td>552</td>
<td>39518.11</td>
<td>71.5908</td>
<td>0.5951</td>
</tr>
<tr>
<td>HP&amp;EDD</td>
<td>559</td>
<td>13545.29</td>
<td>24.2313</td>
<td>0.2014</td>
</tr>
<tr>
<td>SBH</td>
<td>558</td>
<td>12647.81</td>
<td>22.6663</td>
<td>0.1884</td>
</tr>
</tbody>
</table>

In this case, the SBH performs much better than each of the dispatching rule with respect to weighted tardiness related measures. It is interesting to see that the number of completed lots increases significantly by using the SBH approach.

Table 3: Results for the 32 Product Case

<table>
<thead>
<tr>
<th>Rule</th>
<th>#lots</th>
<th>TWT (in h)</th>
<th>AWT (in h)</th>
<th>REL</th>
</tr>
</thead>
<tbody>
<tr>
<td>FIFO</td>
<td>324</td>
<td>452985.26</td>
<td>1398.1027</td>
<td>1.0000</td>
</tr>
<tr>
<td>CR</td>
<td>264</td>
<td>370976.35</td>
<td>1405.2135</td>
<td>1.0051</td>
</tr>
<tr>
<td>HP</td>
<td>338</td>
<td>189682.19</td>
<td>561.1899</td>
<td>0.4014</td>
</tr>
<tr>
<td>EDD</td>
<td>401</td>
<td>418937.61</td>
<td>1044.7322</td>
<td>0.7472</td>
</tr>
<tr>
<td>HP&amp;EDD</td>
<td>412</td>
<td>185340.10</td>
<td>449.8546</td>
<td>0.3218</td>
</tr>
<tr>
<td>SBH</td>
<td>396</td>
<td>89625.57</td>
<td>226.1272</td>
<td>0.1619</td>
</tr>
</tbody>
</table>

The quality of the SBH approach is better in situations where a large amount of lots has a small or medium weight ($D_i$).

In situations with a tight due date setting, the SBH performs slightly better (see Figure 4).

Figure 1 provides results depending on the number of products for the same situation as used before for the results in Table 1 and Table 2.

Figure 2: Impact of an Increasing Number of Products

The quality of the SBH approach with respect to TWT is increasing with the number of products up to 16 products. When we use more than 16 products the quality is slightly decreasing. The same behavior can be observed for the HP & EDD rule. This threshold number of products depends on the used model.

The cycle time of the lots is increasing with the number of products.

In the next three Figures we show the influences of weight setting, due date setting and load of the solution quality. The presented AWT value is an average of the test instances with non-varying factors. Figure 3 show the influence of the weight settings on the AWT values.

Figure 3: Impact of the Weight Setting

The AWT values for different loads are presented in Figure 5.

Figure 4: Impact of the Due Dates

Figure 5: Influences of the Load
The SBH performs in a moderate system better compared to HP&EDD. This is especially true for a large number of products.

Discussion of the Results
From the results in Table 2, Table 3, and Figure 1 we can conclude that the performance of the shifting bottleneck is improved compared to dispatching rules. Furthermore, also the number of completed lots is increased by using the SBH approach in multi-product settings. We did not expect this result. However, it seems that better batching decisions with respect to fullness of batches are the reason for this behavior.

When a certain number of products is reached we do not observe further improvements of the SBH. In the case of many different products, the number of lots that can be batched together is small. Hence, the observed behavior makes sense because non-full batches occur and cannot be improved by better scheduling decisions.

We also show that the SBH clearly outperforms dispatching rules. However, dispatching rules that take the weight and due dates into account perform also quite well. It is clear from Figure 3 that the weights influence the solution quality of SBH. The weights are an important information for the SBH. It is more difficult for the SBH to make good scheduling decisions in case of a large number of lots with the same small weight. Furthermore, tight due dates lead to better results as expected. HP & EDD also performs quite well in case of high loaded systems. However, the SBH performs better in moderate loaded systems.

SUMMARY and OUTLOOK TO FUTURE RESEARCH

In this paper, we studied the influence of the number of products on the performance of a shifting bottleneck heuristic. Therefore, we increased the number of different products in a controlled way. The results show that the number of products has an important influence on the solution quality. Both tardiness and throughput related measures are influenced positively. In future research, we have to perform additional simulation experiments with more complex simulation models. Furthermore, we have to investigate the influence of more sophisticated subproblem solution procedures on the solution quality of the shifting bottleneck heuristic in a multi-product environment. From a practical point of view it seems worth to consider also machine breakdowns and to take rescheduling into account.

REFERENCES


AUTHOR BIOGRAPHIES

JENS ZIMMERMAN is a Ph.D. student in the Department of Mathematics and Computer Science at Fern Universität in Hagen. He received a master’s degree in information systems from the Technical University of Ilmenau, Germany. His research interests are in production control of semiconductor wafer fabrication facilities, multi-agent-systems, simulation, and machine learning. He is a member of GI (German Chapter of the ACM).

LARS MÖNCH is a Professor for Enterprise-wide Software Systems in the Department of Mathematics and Computer Science at Fern Universität in Hagen. He received a master’s degree in applied mathematics, a Ph.D. in the same subject from the University of Göttingen, Germany, and a Habilitation degree in Information Systems from the Technical Universität of Ilmenau. After receiving his Ph.D. he worked for two years for Softlab GmbH in Munich in the area of software development. His current research interests are in information systems for manufacturing, simulation-based production control of semiconductor wafer fabrication facilities, applied optimization, and artificial intelligence applications in manufacturing. He is a member of GI (German Chapter of the ACM), GOR (German Operations Research Society), SCS and INFORMS.
ANALYSIS OF A SEMICONDUCTOR WAFER FAB WITH AN INTERMEDIATE STORAGE FOR PARTIALLY FINISHED LOGIC PRODUCTS

Oliver Rose
Dresden University of Technology
Institute of Applied Computer Science
01062 Dresden, Germany
oliver.rose@tu-dresden.de

KEYWORDS
Manufacturing, Semiconductor, Performance Analysis

ABSTRACT
In semiconductor waferfabs that manufacture both logic and memory products, often intermediate storages are used. From these storages, partially finished logic products are released upon customer orders. In this paper, we discuss some effects of lot priorities and different product mixes on the performance measures of logic and memory products.

INTRODUCTION
In semiconductor industry, mature memory fabs (fabrication facilities) often manufacture a large variety of customer-specific logic products in addition their memory commodity production. It is even the case that they produce a large volume of logic products (make to order) and fill up remaining fab capacity with memory products (make to stock).

In general, about two thirds of the production process for logic wafers is the same for all products and they become customer-specific in the final part. Therefore, several companies have an intermediate storage, sometimes called master stock, for their partially finished logic products. As soon as a customer order comes in, the missing final production steps are performed. This approach should lead to a considerable reduction of the time to market if you compare typical cycle times of about one month to the expected time from leaving the intermediate storage until end of production of about 10 days.

At first glance, the usage of intermediate storages seems to be advantageous but it also brings many problems for the semiconductor fab. The regular fab control tends to avoid large fluctuation in the flow of material. In contrast, the idea of storage as described above behaves completely different if used in a naive way. If a customer orders a special product, the storage releases the wafers and from one moment to another the fab is flooded by a large amount of wafers leading to a high variance process flow, increased inventory (Work In Process, WIP) and cycle time. In addition, it can become difficult to estimate the effect of releasing logic products from the intermediate storage on the other products in the fab. We expect similar problems as mentioned in the literature about wafer surges, e.g. (Dümmler and Rose 2000) or (Rose 2005).

In this paper, we analyze the effects of an intermediate storage for logic products in a wafer fab for a number of different scenarios. For instance, we discuss several release patterns and provide results for different performance measures. We plan to use these results as a basis for the future development of new release strategies for intermediate buffers.

CUSTOMIZING THE SIMULATION MODEL
For our experiments, we used Factory Explorer (FX) Version 2.8.4 from Wright Williams & Kelly. FX does not support the modeling of intermediate storages in fabs. Hence, we had to find a workaround. We divided the production process of logic products into two parts. We called the first two thirds of all process steps of the logic product “Logic raw”. After the “Logic raw” wafer was processed, we started a storage time interval. As soon as the storage period was over the “Logic special” period starts. Each logic lot starts its process as a “Logic raw” product. After 66% of the recipe, it is enqueued as a “Storage product”. The lot waits for the end of setup on the first dummy tool. After a setup period, the lot is put into the queue of the second dummy tool. Upon leaving this tool, it is renamed to “Logic special” and proceeds processing the remaining 33% of the recipe.

REFERENCE RUN
As test model we used the MIMAC (Measurement and Improvement of MANufacturing Capacities) test bed dataset 1. For further details on the datasets and their download: see <www.eas.asu.edu/~masmlab>. The data for this model came originally from a 200nm memory fab. It has two products. We chose product 1 to be the representative for memory products and product 2 to be the representative for logic products. 30% of the production volume are logic products and 70% are memory products. Machine breakdowns are not considered to keep the simulation time acceptable. With breakdowns, a very large number of runs are required to obtain statistically significant results for WIP trajectories over time. We show the effect of breakdowns in the next section. The utilization of the fab is 95%. The dispatch rule is FIFO and no priorities are used. Due dates are not considered.
Every four weeks, all lots of the intermediate storage are released into the fab at once. Figure 1 shows the WIP behavior of the fab model.

![WIP Chart of Reference Run](image1)

Figure 1: WIP Chart of Reference Run

As soon as the storage is emptied, the WIP of “Logic special” rises from zero up to almost 3800 units. After five days, the first special logic wafers reach the end of the production line. About 10 days later, all special logic wafers are finished and the WIP reaches zero again. The number of raw logic wafers in the WIP is almost constant over time (at about 2300 units). The amount of memory products does not show this behavior. Because the special logic product is released with such a high number of units, the fab is practically flooded by logic products. This affects the memory wafers, too, because both products use the same types of tools. The utilization increases and leads blocking. In the chart, the memory WIP curve rises from 7000 units up to 8500 units.

**PRIORITIES**

First, we include priorities into the fab model. In the first scenario, the time critical special logic lots have a low priority (Figure 2).

In comparison to the reference case without priorities (Figure 1), these wafers now need about 22 days to be completed instead of 14 days. But the advantage is that the additional logic wafers have almost no effect on the WIP of the memory products. The memory wafer amount stays approximately constant at a level of about 7000 units.

In Figure 3, the priority effect on memory products is illustrated. If special logic products have a high or no priorities, they have a bad influence on memory products and the WIP rises. It almost makes no difference if there is a high or no priority. Only if special logic products have a low priority the influences on memory products are small. Figure 4 shows the influence of priorities on the completion times of logic wafers. If these wafers have a high priority, the finishing times after storage release are 13 days instead of 14 days. If the priority is low, it takes about 22 days, as seen in Figure 2.

![Invers priorities](image2)

Figure 2: Low Priority for Logic Products

![WIP "Memory"](image3)

Figure 3: Priority Effect on WIP on Memory Products

![WIP "Logic special"](image4)

Figure 4: Priority Effect on WIP of Special Logic Products
PRODUCT MIX

If the product mix changes from 30% logic products to 70% logic products clearly the WIP of logic wafer increases from 2500 to 9000 units for raw logic and from 4000 to 11000 units special logic (Figures 5 and 6). We observe the interesting effect, that, although the memory percentage goes down considerably, the memory WIP stays on a similar level between 6000 and 8000 units. The reason for this behavior might be the larger variation in the material flow due to the logic products. In Figure 5 “only” 4000 units are released from the storage into the fab.

CONCLUSIONS

In this paper, we presented some aspects of a simulation analysis of a semiconductor wafer fab with intermediate storage for logic products. Compared to a factory without such a storage the time to market after a customer order is short. The disadvantage, however, is that production control will be much more difficult. The decrease in time to market goes hand in hand with an increase of inventory for the other product that are not held on stock in the storage. To find a compromise, the planner has to control the number of units released from the storage, the time delay between the released lots, and whether the released lots will have higher or lower priority than the other lots. The effects depend on fab utilization and product mix. Future studies should analyze the effects of dispatch rules, lot release rules like CONWIP or due date strategies to reduce the variance resulting from the storage release. In addition, more studies are needed based on other fab models with a larger number of products.

ACKNOWLEDGMENTS

The author wants to thank Daniel Noack and Falk Pappert for providing the simulation results and for valuable discussions, and Wolfgang Scholl from Infineon Technologies, Dresden, for introducing the planning problems with intermediate storages in wafer fabs to us.

REFERENCES


AUTHOR BIOGRAPHY

OLIVER ROSE holds the Chair for Modeling and Simulation at the Institute of Applied Computer Science of the Dresden University of Technology, Germany. He received an M.S. degree in applied mathematics and a Ph.D. degree in computer science from Würzburg University, Germany. His research focuses on the operational modeling, analysis and material flow control of complex manufacturing facilities, in particular, semiconductor factories. He is a member of IEEE, INFORMS Simulation Society, ASIM, and GI. His web address is www.simulation-dresden.com.
TEXTILE PROCESS SIMULATION
ISSUES IN PROCESSING UNSTABLE TWISTED FIBRE ASSEMBLIES

Canh-Dung Tran
David G. Phillips
CSIRO Textile and Fibre Technology
PO Box 21 Belmont, Victoria 3216, Australia
Canh-dung.tran@csiro.au
David.phillips@csiro.au

G.H.M. van der Heijden
Centre for Nonlinear Dynamics
Civil Engineering Building UCL
London, United Kingdom
ugesvdj@ucl.ac.uk

KEYWORDS
Topological conservation law, writhe, twist, link, multi-ply, balance twisted yarn, instability.

ABSTRACT
During yarn formation by ring spinning, fibres are bent into approximately helical shapes and an unbalanced torque or twist-liveliness is created. Generally, the torque depends on yarn geometric factors such as the yarn twist, linear density and the fibre properties. In order to prevent the instability of yarn (e.g., snarling), twist-liveliness must be controlled. A practical solution to the problem of twist-liveliness is the formation of a two-fold yarn.

In the present paper, the instability of twisted textile yarns will be interpreted with a model built using the Topological Conservation Law (Fuller, 1971) which has been developed to study the dynamics of twisted rods by Van der Heijden et al. (2003). The present work considers the equilibrium configuration of a series of multi-ply twisted yarns (2, 4, 8, and 16 strands) of finite length. The influence of structural properties (the number of strands, the diameter and twist of each strand) on the parameters of a balanced multi-ply yarn (writhe and twist) using a topological invariant of the twisted yarn (link) is established and investigated using experimental results obtained at CSIRO.

INTRODUCTION
For the last decade, one of the important subjects of the dynamics of twisted yarns is the source of their instability. In fact, issues of instability for twisted structures are well-known, such as snarling of yarn or the skewing of knitted fabrics. For example, if the ends of a freshly-spun yarn are brought together, the yarn will buckle locally and jump into a ply or form a snarl. By assuming helical shapes for the centre-line of strands in the ply, expressions of forces and moments in the ply can be obtained using the static equilibrium equations. Recently, based on the topological conservation law (TCL) (Fuller, 1971, 1978) for the dynamics of rods, Thompson et al. (2002) has employed the dynamics of a elastic fibre to model a single DNA molecule and analysed its super-coiled equilibrium configurations. Van der Heijden et al. (2003) have modeled the non-linear jump behaviour of twisted clamped rods. In textile engineering, the twisting or plying process produces a yarn structure where the energy of the system is determined by purely geometrical constraints of the plied structure and consequently when an energy minimum is reached the plied yarn obtained from the process is torsionally balanced and torque-free. The instability of such yarn structures will be considered using this law (TCL) in the present work. The equilibrium configuration of a series of multi-ply twisted yarns of finite length and some influences of structural properties on a balanced multi-ply yarn (writhe and twist) are investigated and presented.

MODEL OF A MULTI-PLY YARN
One way of representing the conformation of a strand within a multi-strand yarn is to regard the strand as lying on the surface of a cylinder. Figure 1 shows the case of a yarn of n strands of radius r whose centrelines are wound on a cylinder of R, where each strand is considered as an elastic circular filament. Hence, the configuration of a strand is specified by the position of a curve in space r(s) where s is the arc length along the central axis of the yarn.

Let \{X_1, X_2, X_3\} be a fixed rectangular Cartesian coordinate system, and \{x_1, x_2, x_3\} a moving coordinate system whose two axes coincide with the principal axes of the strand cross section (x_1 and x_2) and the third axis x_3 coincides with the tangent to the centre line of the strand \( \frac{dr}{ds} \) (see Figure 1). Let \( \psi, \theta, \phi \) be the angular twist in the strand, the helical angle inclination of the strand and the angular rotation of the strand around the x_3 axis, respectively.

Figure 1: Schema of a strand (radius r) wound on a cylinder (radius R) with the fixed cartesian and moving coordinate frames.
For the sake of presentation, considering strands wound on a cylinder in the left handed helix as shown in Figure 1. The centreline of a strand \( (i) \) can be expressed in fixed frame (Neukirch and Van der Heijden, 2002, Van der Heijden, 2001) as follows

\[
\mathbf{r}_i(s) = K_{i}^{a_{i}}\mathbf{r}_i
\]

(1a)

With

\[
K_{i} = \begin{pmatrix}
\frac{\cos 2\pi}{n} & -\frac{\sin 2\pi}{n} & 0 \\
\frac{-\sin 2\pi}{n} & \frac{\cos 2\pi}{n} & 0 \\
0 & 0 & 1
\end{pmatrix}
\]

and \( \mathbf{r}_i = \begin{pmatrix} -R \sin \psi \\ R \cos \psi \\ \cos \theta \end{pmatrix} \)

(1b)

Where \( \psi(s) = \frac{\sin \theta}{R}, \quad \psi(0) = 0 \)

(1c)

Hence, the position vector \( \mathbf{r}_{Ai} \) of an arbitrary point \( A \) (see Figure 1) on the surface of a strand \( i \) can be determined by

\[
\mathbf{r}_{Ai}(s,t) = \mathbf{r}(s) + \cos \phi \mathbf{e}_x + \sin \phi \mathbf{e}_y
\]

(2)

The geometrical model considered here is based on elastic rod mechanics with the assumption that the distance of the centreline of two adjacent strands is minimum and equal to the diameter of each strand. This is expressed as follows:

\[
2 + m^2 \cos^2 \theta - 2 \cos \left( m \sin \theta - \frac{2\pi}{n} \right) = \frac{4}{\rho^2}
\]

(3a)

\[
m \cos^2 \theta + \sin \theta \sin \left( m \sin \theta - \frac{2\pi}{n} \right) = 0
\]

(3b)

Where \( m = \frac{S_{i+1} - S_i}{R}, \quad \rho = \frac{R}{r} \)

Equations (1b,c) and \( \mathbf{x}_i = \frac{d\mathbf{r}}{ds} \) yields to

\[
\mathbf{x}_i = \begin{pmatrix} \sin \theta \cos \psi \\ \sin \theta \sin \psi \\ \cos \theta \end{pmatrix}
\]

(4)

Let \( u_i \), whose components are \( u_{i1}, u_{i2}, u_{i3} \), be the curvatures and the twist of strand in the moving frame, the governing equations of the evolution of a strand (the centreline) are given by

\[
\frac{dx_{i1}}{ds} = u \times \mathbf{x}_i, \quad i = 1, 2, 3
\]

(5)

Because the centreline of each strand lies on a cylinder of radius \( R \), a cylindrical coordinate frame \( \{ \mathbf{e}_r, \mathbf{e}_\theta, \mathbf{e}_z \} \) is introduced as follows

\[
\mathbf{e}_r = -\sin \psi \mathbf{X}_r - \cos \psi \mathbf{X}_z;
\]

\[
\mathbf{e}_\theta = -\cos \psi \mathbf{X}_r + \sin \psi \mathbf{X}_z;
\]

\[
\mathbf{e}_z = \mathbf{X}_z
\]

(6a)

With

\[
\frac{d\mathbf{e}_r}{ds} = \frac{d\psi}{ds} \mathbf{e}_r; \quad \frac{d\mathbf{e}_\theta}{ds} = -\frac{d\psi}{ds} \mathbf{e}_r; \quad \mathbf{x}_i = -\sin \theta \mathbf{e}_\theta + \cos \theta \mathbf{e}_r
\]

(6b)

**GOVERNING EQUATIONS OF A MULTI-PLY YARN**

**Link, Twist and Writhe in Textile Engineering (Kinematic Equation)**

For this presentation, consider a single Z twisted yarn of length \( L \), whose ends are glued together to form a closed loop or hank. Let the number of total end rotations of the yarn, before being formed into the hank, be specified as the link which is a topological invariant, i.e. it is unaffected by any subsequent deformation and is defined for as the following formula (Van der Heijden et al., 2003 and Fuller, 1971)

\[
L_z = T_x + W_t
\]

(7)

Where \( T_x = \frac{1}{2\pi L} \int u_i ds \) is the total twist, and \( W_t \) (named the writhe) is a property of the centreline of the single yarn. It is produced from the out-of-plane deformation to form a snarl or plied yarn and equal to the signed crossing number averaged over planar projections from all possible directions (see Thompson et al., 2002, p.963). In a spinning process, the concept of the “twist” that is inserted into a single twisted yarn, is actually the ‘link’ per unit length of yarn. If applying a rotation of \( \Omega \) (rads) in a single yarn of length \( L \), the spinning twist per length \( \tau \) (positive for Z twist and negative for S twist) and twist angle are given by, respectively

\[
\tau = \frac{\Omega}{L}, \quad \tan \theta = \tau t.
\]

(8)

Since the twist is assumed constant along the yarn in textile engineering, the link is written as follows

\[
L_z = \frac{\Omega}{2\pi} \equiv \frac{L_z}{2\pi}
\]

(9)

A similar explanation holds for a multi-ply assembly from more than one hank and the details can be seen in Neukirch and Van der Heijden (2002).
Kinetic Balance Equations

Consider a multi-ply yarn of \( n \) strands whose physical and geometrical parameters are mentioned in previous sections. The governing equations of a strand for the force \( \mathbf{F} \) and moment \( \mathbf{M} \) are given by (Fraser and Stump, 1998a,b; Thompson et al., 2002; Stump and van der Heijden, 1999)

\[
\frac{d\mathbf{F}}{ds} + p = 0 \tag{10a}
\]

\[
\frac{d\mathbf{M}}{ds} + \frac{dr}{ds} \times \mathbf{F} = 0 \tag{10b}
\]

Where \( p \) is the pressure (force per unit length) on a strand in which the constitutive relations between the stresses and strains are assumed linear and given by

\[
\mathbf{M} = B x_3 \times \frac{dx_3}{ds} + C u_3 x_3 \tag{11}
\]

and \( B \) and \( C \) are the bending stiffness and torsional rigidity of the strands, respectively. Let \( (F_x, F_y, F_z) \) be the components of \( \mathbf{F} \) in the cylindrical coordinate, we have

\[
\mathbf{F} \times \mathbf{d}_z = (F_x \cos \theta + F_y \sin \theta) \mathbf{e}_r - F_y \cos \theta \mathbf{e}_r - F_x \sin \theta \mathbf{e}_r \tag{12}
\]

From (12) and (11), the differential balance Equation (10b) is rewritten as follows

\[
-B \frac{d\theta}{ds} + B \left( \frac{d\psi}{ds} \right) \sin \theta \cos \theta - C u_3 \frac{d\psi}{ds} \sin \theta = -F_y \cos \theta - F_x \sin \theta \tag{13}
\]

\[
F_x = -B \frac{d\psi}{ds} \sin \theta - 2B \frac{d\psi}{ds} \frac{d\theta}{ds} \cos \theta + C u_3 \frac{d\psi}{ds} \tag{14}
\]

\[
C \frac{du_3}{ds} = 0 \tag{15}
\]

The derivatives of \( \mathbf{F} \) in the cylindrical frame are given by

\[
\frac{dF_x}{ds} = -\frac{F_x}{R} \sin \theta - p; \quad \frac{dF_y}{ds} = \frac{F_y}{R} \sin \theta; \quad \frac{dF_z}{ds} = 0 \tag{16}
\]

With the assumption \( \theta \) and \( u_3 \) are constant, Equations (14), (16) and (11) show that the components of \( \mathbf{F} \) are all constant and are calculated as follows

\[
F_x = 0
\]

\[
F_z = \frac{F_z}{n}
\]

\[
F_y = \frac{M_\theta}{nR} + \frac{B}{R} \sin \theta - \frac{C u_3}{R} \cos \theta
\]

where \( F_x \) and \( M_\theta \) are the axial force and moment applied to the multi-ply yarn, respectively. From (17), the differential balance Equation (13) is rewritten by

\[
2n \sin^3 \theta \cos \theta - n \frac{C}{B} \rho u_3 \cos 20 + \frac{\rho\gamma}{L} \frac{dF_x}{ds} - \rho M_\theta \cos \theta = 0 \tag{18}
\]

With a given initial twist \( u_3 \), it can be expressed as follows (Antman, 1995, Love, 1904)

\[
u_3 = \frac{d\psi}{ds} + \frac{du_3}{ds} \left( \cos \theta - \frac{\sin \theta}{L} \right) = \frac{\gamma}{\rho} - \frac{\sin \theta}{2R}
\]

Substituting (19) into (18) yields

\[
2n \sin^3 \theta \cos \theta + \frac{C}{B} \rho \sin 40 \gamma - \frac{\rho\gamma}{L} \frac{dF_x}{ds} + \frac{\rho M_\theta}{B} \cos \theta = 0 \tag{20}
\]

The equation kinetic (20) describes the relationship of structural parameters \((n, \theta, C, B, r, L, \tau, R)\) and dynamic parameters \((F_i, M_i)\) of a multi ply yarn at the balance situation.

The Balance of a Multi-Ply Yarn

After removal of the applied force and moment \((F_x = M_\theta = 0)\), the multi-ply yarn reaches the balanced state and the Equation (20) becomes

\[
2n \sin^3 \theta \cos \theta + \frac{1}{4} \gamma \sin 40 - \frac{\gamma}{L} \rho \cos 20 = 0 \tag{21}
\]

where \( \gamma = \frac{C}{B} \). Equation (21) gives the relationship between twist and writhe of the process of transformation from the single twisted yarn to \( n \)-ply yarn. Hence, with given yarn structural parameters (yarn count, twist, bending stiffness, torsional stiffness and the number of strands of a multi-ply yarn), the balance state described by the helical angle \( \theta \) of the multi-ply yarn is totally determined by (3) and (21).

EXAMPLES AND DISCUSSIONS

In this section, the balance of multi-ply yarns comprising 2, 4, 8, 16 strands will be analysed. The analysis is carried out using a range of different yarn counts and initial twists in which the relationships between twist angle \( \gamma \) of strands and helical angle of the multi-ply yarn (or the corresponding quantity) are determined. Although the torsional and bending stiffnesses of strands depend on the structural parameters of the yarn, the ratio of the torsional and bending stiffness \( \psi \) and the Poisson’s ratio \( \nu \) are assumed constant at about 0.7 and 0.43, respectively, in the present work. Furthermore, the radius of wool worsted slants is calculated as a function of the yarn count using the packing fraction of 0.63 by Booth, (1975).
Plies From One Hank

For plied twisted yarns from one hank, i.e., two strands (n = 2, R = r (p = 1)): this is a common and simple case of multi-ply yarns. In the balanced state, the kinematic and balance kinetic Equations (7) and (21) yields

\[ \tan \phi_x = \tau_{EF} = \frac{\sin \theta}{r} \left( \sin \theta \tan 2\theta + \gamma \cos \theta \right) = \frac{-L}{2(R - 2 \gamma \sin^2 \theta)} \]  

(22)

where \( \phi_x \) is the twist angle of the two strands and \( \gamma = \frac{(1 - \gamma^2)}{\gamma} \) is the Poisson’s ratio of yarns. The link and write can be determined from (7), (8) and (22).

\[ L = \frac{L}{2 \pi r} \left( \gamma \sin \frac{2\beta}{1 + \gamma} \tan 2\beta \right) \]

\[ W = \frac{L}{2 \pi r} \sin 2\beta \]

Figure 2 The ratio (%) of helix angle of the poly and twist angle of strand plotted against the twist angle of strand using \( \gamma \)'s.

Figure 3 The ratio (%) of write of the balanced yarn to the link plotted against the link using different yarn.

against the strand twist angle in Figure 2. The ratio of write (ply twist) to the link is plotted against the link in Figure 3. The data in Figure 3 is expressed in terms of yarn count but this reflects the effect of the yarn diameter which is a geometrical property of strands, i.e., yarn count and packing fraction. The present analysis using the TCL and the kinetic balance point is compared with preliminary experimental results for a series of two-plied yarns using 80 tex singles ranging from 60 to 140 metric twist factor (Figure 4). The difference between the theoretical values and the experimental results is always less than 8% for two strand plied yarns.

Figure 4 The ratio (%) of Write of the balanced 2-ply yarn to the Link plotted against the Link showing theoretical and experimental results for 80 Tex strands.

Figure 5 The helix twist angle of a balanced 4-ply yarn plotted against the twist of strands for three yarn counts; 80 Tex, 100 Tex and 140 Tex.

Plies From More Than One Hank

The influence of yarn count and the number of strands on the balance conditions for a multi-ply n-strand yarn is considered in the following. Figure 5 shows the theoretical relationship between twist angle of strands and the helical angle of the multi-ply yarn of strands with respect to yarn count (80 Tex, 100 Tex, 140 Tex) at the balance situation. The relationship in Figure 6 shows the effect of an increased number of strands from 4 to 16.
Figure 6 The theoretical relationship between the helix twist angle of multi-plyed yarns plotted against the twist of strands for balanced plies of 4, 8, 16 strands of 80 Tex yarn.

Due to the limitation of paper length, some other results which do not appear in the paper will be presented at the conference.

CONCLUSION

The Topological Conservation Law is used in combination with the governing equations to simulate the equilibrium configuration of a balanced multi-ply yarn. The influence of structural geometry on balance conditions in the multi-ply yarn is also established and investigated using preliminary experimental results obtained at CSIRO. The relationship of the link and writhe of a given closed yarn structure may allow multi-ply twisted yarns to be designed whose instability (balance) can be controlled.

ACKNOWLEDGEMENTS

This work was supported by a grant of computing time from Australia Partnership for Advanced Computing (APAC) National Facility. Dr Canh-Dung Tran is supported by a CSIRO Postdoctoral Research Fellowship. The experimental data on balance in multi-ply structures was obtained through a CSIRO Textile and Fibre Technology project. All of this support is gratefully acknowledged.

REFERENCES


COMPOSITE TEXTILE STRUCTURE FOR BALLISTIC PROTECTION

François BOUSSU, Virginie BEGUS
GEMTEX Laboratory (Genious and TExtile Materials),
ENSAIT (French Engineering School in Textile),
9 rue de l’Ermitage, BP 30329,
F-59056 Roubaix Cedex France.
francois.boussu@ensait.fr, virginie.begus@ensait.fr

Jean-Luc PETITNIOT
ONERA (National Office of Research and Studies in Aeronautics)
DMSE/RCS
5, boulevard Paul Painlevé
F-59045 Lille France.
Jean-Luc.Petitniot@onera.fr

KEYWORDS
Military Applications, Approximation technique,
Mechanical engineering.

ABSTRACT

In this paper, an investigation has been conducted to
determine the different phenomenon which occurred during
a high or low velocity impact. These phenomena can be
mainly explained by different type of mechanical stresses,
the temperature spreading of the projectile inside the
composite structure and kinetic energy absorbed by the
material. Considering these main parameters, the design of
the composite structure has been oriented to integrate
different decreasing hypothesis of the kinetic energy
absorbed and temperature spreading. Different textile
solutions are proposed to the design of a new body amour.

INTRODUCTION

First, let’s have a look at how a Body Armour works during
the impact of a handgun bullet?

Very strong fibres absorb and disperse the impact energy
that is transmitted to the vest from the bullet, causing
the bullet to deform. Additional energy is absorbed by each
successive layer of material in the vest, until such time as
the bullet has been stopped. Because the fibres work
together both in the individual layer and with other layers
of material in the vest, a large area of the garment becomes
involved in preventing the bullet from penetrating. This
also helps in dissipating the forces which can cause non
penetrating injuries (what is commonly referred to as “blunt
trauma”) to internal organs. Unfortunately, at this time no
material exists (considering the state of the art available for
the public) that would allow a vest to be constructed from a
single ply of material.

Currently, today’s modern generation of concealable body
armour can provide protection in a variety of levels
designed to defeat most common low- and medium-energy
handgun rounds. Body armour designed to defeat rifle fire
is of either semi rigid or rigid construction, typically

incorporating hard materials such as ceramics and metals.
Because of its weight and bulkiness, it is impractical for
routine use by uniformed patrol officers and is reserved for
use in tactical situations where it is worn externally for
short periods of time when confronted with higher level
threats.

Since 2000, a GEMTEX laboratory team is engaged in a
campaign to develop new armour including new kind of
high performance yarns. As a matter of fact, assuming a
certain number of hypotheses and considering elementary
computations on the buckling model of fabrics to impact, a
new solution is proposed and will be soon tested.

The new backing will have to integrate different fabrics
with different type of yarns where each of them is used for
its main property during the impact. Its main composition is
presented in the

Figure 1:

![Figure 1: Description of the future backing structure to test.](image)

The main awaiting properties of the four main blocks of the
backing structure are detailed in

Table 1. Thanks to our previous hypothesis, this new
backing will succeed at the different level tests and hope to
reach a rear back buckling quasi null for the comfort of the
human body.

New recommendations as regards the different types of
yarns use are discussed.
Table 1. Awaiting functions of the different fabrics inside the backing.

<table>
<thead>
<tr>
<th>Number of plies</th>
<th>Description</th>
<th>Awaiting function</th>
</tr>
</thead>
<tbody>
<tr>
<td>N1 plies</td>
<td>Fabrics 1 and 2 made of PEHT yarns (Dyneema ™) with respectively a higher linear density for the fabric 1 than the fabric 2. Each of these fabrics has a specific weave diagram in order to be well coupled.</td>
<td>Decreasing of the shearing stress and absorption of the main part of energy during the projectile penetration.</td>
</tr>
<tr>
<td>N2 plies</td>
<td>Fabric 3 made of polypropylene with high linear density of yarns.</td>
<td>Absorption of the calorific energy by melting</td>
</tr>
<tr>
<td>N3 plies</td>
<td>Fabric 4 and 5 made of high tenacity aramid (Technora ™) with respectively a higher linear density for the fabric 4 than the fabric 5. Each of these fabrics has a specific weave diagram in order to be well fitted.</td>
<td>Decreasing of the shearing stress</td>
</tr>
<tr>
<td>N4 plies</td>
<td>Fabric 6 made of shape memory alloy wires and aramid yarns</td>
<td>In the transition zone of the SMA, damping capacity at the impact.</td>
</tr>
</tbody>
</table>

USE OF PEHT YARNS IN BALLISTIC

As mention in the web site of DSM, the Dyneema® UD provides protection against all major threats due to its ultra-high energy absorption. The UD construction makes backing deformation wider and less deep, thereby minimizing blunt trauma effect. UD stands for ‘unidirectional’: in Dyneema® UD all the fibres are laid parallel, in the same plane, rather than being woven together. Dyneema® UD is made of several layers of Dyneema® fibres, with the direction of fibres in each layer at 90° to the direction of the fibres in the adjacent layers. The unidirectional configuration of the fibres in Dyneema® UD allows the energy transferred from the impact of a bullet or other threat to be distributed along the fibres much faster and more efficiently than in conventional woven fabrics. This is because the absorption power of the yarn in woven fabrics is lost at the crossover points, as these points reflect rather than absorb the shockwaves of the impact. Thus, we recommend using non conventional woven fabrics which allow keeping the warp and weft yarns as straight as possible into a specific weave diagram. For our solution, the woven structure is kept for its convenience as an “easy to use” material and the number of crossover points will be minimized. An optimization has to be done on the different ends and picks densities to use for the different fabrics to be joined. The main idea is to insert crossover points of the first fabric inside the “hole” of the second fabric and reversely.

The family of Spectra® Fibres (introduced by Honeywell) provides higher strength, modulus and improved creep properties as well as fibres with enhanced ballistic performance in armour. As an extended chain, ultra-high molecular weight polyethylene, Spectra® 1000 fibres has a high strength-to-weight ratio. Thus, the design of our new textile composite structure puts on the fore the polyethylene fibres inside a fabric with a specific weave diagram.

USE OF ARAMID YARNS IN BALLISTIC

Civilian police forces that are exposed to relatively high-mass/low-velocity bullets will use a different fibre and different fabric weave designs than military applications which encounter relatively low-mass/high-velocity bullets and fragments. Lightweight, durable and comfortable material is vital in this industry to make sure that the users will not hesitate to wear soft body armour for protection. As regard the web site of Dupont, Kevlar® fibre displays excellent dimensional stability over a wide range of temperatures for prolonged periods. Even at temperatures of 160° C Kevlar® shows essentially no strength loss. Kevlar® fibre also has excellent dimensional stability with a slightly negative coefficient of thermal expansion (-0.2 x 10^-6°C). Kevlar® 29 & 129, a tougher yarn, is more suited for ballistic protection and is used both in soft body armour applications such as personal protection ballistic vest, and as a robust reinforcement for a variety of hard armour applications as ballistic helmets. These yarns are used in both soft and hard military products, to defeat multiple threats. Considering the information of the web site of Teijin, developments of finer filament count, uniquely combined with higher tensile properties, have led to the newest types of Twaron®, exceptionally well adapted to the specific demands of ballistic applications both hard and soft. The main interesting property of these fibres lies in their mechanical stability under a range of temperature from 120 to 180° C.

USE OF SHAPE MEMORY ALLOW WIRES IN BALLISTIC

Smart fabrics, including special yarns as shape memory alloy wires, have been realized in the GEMTEX laboratory. A weaving technique, unique in the world as mention in (Arata et al. 2004), has been developed (Boussu et al. 2001). Special requirements during the warping, drawing-in and weaving processes have to be taken in order to keep the material into an austenitic phase. Thus, thanks to the obtain fabric of 100% Nitinol SMA, as well as in the warp and weft directions, different properties has been shown with respect to damping capability and the super elastic effect (Boussu and Petiti6n 2002). It follows from this that several experiments have to be achieved with SMA fabrics coupled with poly(paraphenyleneterephthalamide (PPTA) and high tenacity polyethylene (PE) fabrics to make a composite structure improving the high velocity impact resistance. In this patent, dealing with methods of protecting structures from impacts, the components are interposed between a
point of impact and a structure to be protected (Cuniff 1992). They comprise a SMA exhibiting pseudo-elastic behavior, and having a high strain to failure. By the same, in this publication, experimental results on the dynamics of a beam constrained by shape memory wires are presented (Paine and Rogers 1994). It is observed that the damping increases significantly when the shape memory wires are stressed such that they lie within the pseudo elastic hysteresis loop. These results demonstrate that pseudo-elasticity of shape memory wires can be used to augment passive damping significantly in structural systems. This indicates that the SMA yarns to be used in our ballistic application have to be in a transition phase depending on the stress and the temperature. The super elastic SMA is shown to be effective at low velocities and may be also in high and ballistic velocity applications. A previous study has been achieved by Kiesling and it was demonstrated that an increase of 41% of the energy absorption can be obtained only with 6% of SMA inserted in volume (Kiesling et al. 1996). Thus, an adjusted proportion of SMA fabrics will be used in our ballistic application with respect to the total volume and weight. At last, in the thesis work of Roger Ellis (Roger 1996), just after Paine and Kiesling (Paine and Roger 1995), the concept of using high strain SMA and ECPE hybrid components to improve the ballistic impact resistance of graphite composites has been studied. The following obtained results have to be highlighted:

- A relative improvement of 99% of the energy absorption is observed when the Spectra yarns are located to the back of the composite with only an increase of 12% of the total weight.
- Other and pure SMA fabrics must be tested by varying the yarns diameters and the alloy compositions.

Finally, by taking into account all these previous results and recommendations, different backings have been realised made of pure aramid and blend of aramid and SMA fabrics corresponding to the NIJ norm standard level 3 and 4. At the level 3, the armour protects against 7,62 mm full metal jacket bullets (US military designation M80), with nominal masses of 9,7 g impacting at a velocity of 838 m per second or less. Projectiles are fired six times at different located aims. All the armours have been manufactured by MS Composite Company, during a student final year project of six months. At the level 3, all of the armours succeed at the ballistic tests. The mean resulted deformation after the impact for all these tests was half time very low. At the level IV, the armour protects against 30 calibre armour-piercing bullets (US military designation APM2), with nominal masses of 10,8 g impacting at a velocity of 868 m per second or less. Only one of five armours tested was failed at the level IV, mainly due to an excessive velocity of the projectile (measured at 890 m/s instead of 868 m/s).

CONCLUSION

In the ballistic domain, different ways put on the fore the main interest to make a blend of different yarns inside the composite structure to take the benefit of each of their thermo-mechanical properties. Different tests have allowed validating a certain number of hypotheses as regards the yarns behaviour at high velocity impacts and specifically the SMA wires. The obtained results are sufficient but can be improved by varying a certain number of parameters to aim at: a final composite structure solution optimizing a criterion including the total cost and weight.

REFERENCES


AUTHOR BIOGRAPHY

FRANCOIS BOUSSU was born in Amiens, France and went to ENSAIT (Textile Engineer School) to obtain his degrees in 1992. He worked, during his military service, for two years in an embroidery factory in Tunisia. He became head of production and computer supervisor for another lace production factory in France for one year. He started a Ph-D work at the University of Lille in 1995 and obtained its diploma in 1998. Since 1998, he is an assistant professor at ENSAIT, specialized in the warp and weft structure, and its main research works, at the GEMTEX laboratory, deals with the new development of fabrics for technical textiles.
DIAGNOSIS, THERMAL AND HYDRODYNAMICAL MODELLING IN BOBBIN DYEING PROCESSES

Carlo Beltramo and Giorgio Rovero
Department of Chemical Engineering and Material Science
Politecnico di Torino
Corso Duca degli Abruzzi 24
10129 Torino, Italy
E-mail: carlo.beltramo@polito.it

KEYWORDS
Yarn dyeing, pressure drop, system permeability, dyeing diagnostics, bobbin temperature mapping.

ABSTRACT
A diagnostic approach to identify pressure drop distribution and temperature profiles in yarn packages is presented. A two bobbin dyeing apparatus, suitable for hydrodynamic measurements, was used by inserting thin pressure probes into the yarn structure. Pressure drop profiles were interpreted according to calibration curves generated in cylindrical radial elements of the package by relating to the imposed flow rate. The permeability of the fibrous element was varied to cover the density measured during standard bobbin winding. Hydrodynamic modelling is provided to contribute to the interpretation of hydraulics both with internal to external and with external to internal direction of flow. Permeability and pressure drop tests were run at various temperature. The two-bobbin dyeing pilot plant apparatus was also used to map the temperature profiles inside the yarn package according to a temperature raise strategy typical of usual dyeing operations. Noticeable temperature gradients were recorded and related to the sensitivity of actual dyeing kinetics to this physical variable. Fundamental considerations on the hydrodynamics of the dye bath inside a fibre package were obtained by imposing a temperature step to the equipment, thus devising the characteristics of heat and mass transfer from the liquor to the textile material. A model of the system was conceived by analysing the temperature profiles and by separately describing the flow inside the packages and in the other equipment components.

INTRODUCTION
Yarn package dyeing is an extensively used process in textile industries thanks to its easy handling and an highly automated production. The clue point, as a result of the entire process, is an even colonization of the yarn in the package, since the product is expected to have a high quality standard. In the view of minimizing water and chemical consumption, the liquor ratio in the equipment has been progressively reduced to pursue a technology enhancement perspective. On the other hand, a reduced liquor ratio leads the process to work closer to the limiting physicochemical conditions which take place in the dye-fibre system. The entity of not-evenly dyed yarn in the package may reach 4 to 6% of the mass, then implying an additional cost due to product re-dyeing. Starting from this technological considerations, the need for a more comprehensive understanding of yarn package dyeing follows, to include hydrodynamic and thermal phenomena, beside consolidated physicochemical interpretations. The large number of operative parameters and their complex interconnection need to be related by a diagnostic analysis, which may usefully enhance criteria of dyeing management. This paper presents the results of a methodological approach to investigate the possible hydrodynamic and thermal causes of uneven colorations. The incomplete knowledge both on transport phenomena and hydrodynamics in a fibrous structure suggested to undertake this research. The ultimate goal of this combined research project, aimed at the identification of hydrodynamics and thermal mapping in yarn packages, has consisted in relating these fundamental parameters to coloration unevenness, which might be manifested at the end of the dyeing process in the yarn layers closest to the bobbin former or where liquor bypass had occurred.

EXPERIMENTATION
As a first step, the experimentation aimed at relating pressure drop profiles versus fluid velocity through a fibrous bed. This result was obtained by coring out an element of the package along the radial direction to carry out specific permeability tests according to an imposed flow pattern. The second step of the experimentation on yarn packages was intended to directly generate values of liquor pressure drop along the bobbin radius. By relating the results obtained in the experimental sequence, the liquor velocity field was mapped along the yarn package radius and axis. Additionally, pressure profiles provided a clear evidence of a non-negligible fluid bypass towards bobbin interface.

The third step of this research involved the thermal experimentation on yarn bobbins during each phase of the dyeing process. In fact, one of the most important physical variable that influences color levelness may be given by temperature gradients in a bobbin, as a consequence of the
thermal dynamics imposed by the control system. Temperature regulates the chemical dyeing equilibrium and the dye diffusion in fibres. Besides, some materials (acrylic is a typical example) pass through glass transition in a given temperature interval. This phenomenon causes fibre softening, which on its turn affects hydrodynamics and liquor flow distribution in the package, by altering its structural characteristics (porosity and permeability).

To summarize the experimental procedure of the project can be schematically divided into:

1. permeability measurements on fibrous elements,
2. pressure drop mapping along several radial directions in a bobbin
3. temperature ramp test
4. temperature step test

The results of the first two steps were then related to obtain the liquid velocity field in the yarn package. Some non-uniformity, as far as the interaction between dyeing liquor and fibres is concerned (bypass and dead zones in certain zones of the package - by the former or at the interface between adjacent bobbins) was detected. The yarn packages considered for the measurements had a compressible polypropylene former (Microcot type), an internal diameter of 64 mm, an outside diameter of 220 mm, a height of 165 mm and an average weight of 2.00 kg.

**Permeability Evaluation**

To accomplish the first step, a yarn package was impregnated by warm liquid paraffin in order to maintain unchanged the fibre structure during the subsequent extraction of a yarn specimen carried out on the solidified structure. In this operation a cylindrical element was cored out from the embedded package along a radial direction, according to Figure 1.

![Figure 1: Extraction of a Solidified Yarn Specimen from a Bobbin](image)

The specimen was positioned into a pressure drop test apparatus (Figure 2), where the paraffin was thoroughly removed by n-heptane extraction. Other specimens were obtained by coring water-frozen packages to further simplify the yarn element preparation.

Water was flown through the fibrous bed at controlled rate and temperature: static pressure drop profiles were measured by 0.6 mm O.D. capillary probes inserted perpendicularly with respect to the flow direction. Tests were run in the bed of fibres at a different porosity by imposing a proper axial. The value of permeability measured on specimens prepared with both coring techniques were matching.

![Figure 2: Pressure Drop Test Apparatus](image)

**Pressure Drop Measurement**

A pilot plant designed to hold two yarn packages was used in the second step of this program. This equipment represents a short element of a tubular dyeing machine produced by OBEM S.p.A.. Typical operative conditions for flow rate, temperature and bobbin compression were reproduced in the pilot unit.

The equipment main parts are: a dyeing autoclave (25 L volume, R/B = 1/6), suitable to hold two piled packages, a centrifugal pump to induce dye-liquor circulation in radial direction through the bobbins, a 4-way valve to reverse the liquor flow from the I/E (inside-to-out direction) to the E/I operation and a control panel to display and regulate the essential circuit variables (flow, temperature, pressure drop through the bobbins). The pilot plant configuration for pressure drop measurement is given in Figure 3. The pressure signals were sent to the pressure transducer via a 24-way valve: steady state reading was attained after an about two minute transient upon probe channel commuting. Several packages were wound by inserting capillary probes according to a space distribution map which implied four radial and three axial positions. Pressure drop profiles at top, mean and bottom locations were monitored. This approach offered the possibility of enlightening the internal hydrodynamics of a yarn package, thus demonstrating uneven distribution of the dyeing liquor due to density variations originated during bobbin preparation and to compession exerted along the spindle assembly.
Temperature Test Apparatus

The pilot plant of Figure 3 was also used for an experimental study on temperature profiles, after the necessary modifications. As shown in Figure 4 the pilot plant was equipped of a 140 L heated tank and a 8 kW heating element to implement heating gradients in the dye bath matching production conditions. Additionally, a data logger for temperature data acquisition was installed.

![Figure 3: Pressure Drop Test Pilot Equipment](image)

Bobbin preparation required the insertion of 24 1.2 mm O.D. stainless steel capillary tubes among the yarn layers during winding; similarly to pressure probe radial positioning, distances of 2, 9, 14, 25, 50 and 67 mm from the package former were chosen. After winding, each tubing was withdrawn and replaced by a 1.0 mm thermocouple featured by a high accuracy calibration. Two further thermocouples were positioned inside the autoclave: one was inserted into the dyeing spindle and another one was placed at the outer surface by the two bobbin interface (see Figure 5). All the thermocouples were connected to the data logger to on-line record the temperature profile dynamics.

![Figure 5: Thermocouple Disposition and Fluid By-pass](image)

Two types of stimulus-response tests (Levenspiel, 1962) were carried out: a ramp and a step function were chosen to analyse the system reaction to heat capacity alterations. The ramp test consisted in imposing a constant temperature gradient (0.8 or 1 °C/min) to simulate a typical temperature raise in a dyeing process. The liquor heated in the external tank was flown into the kier through the two piled bobbins and returned to the heating tank to further increase its temperature. The step test consisted in transferring a given volume of hot fluid from the external heated tank to the bobbin-loaded autoclave, which was previously operated at a lower steady temperature. This test aimed at quantifying the time delay dynamics due to the liquor-to-yarn heat transfer parameters (heat capacity, heat transfer coefficient and property dispersion along the fluid direction).

RESULTS

Permeability Evaluation

A usual indirect method to detect the actual fluid velocity in a porous arrangement consists in mapping pressure gradients in the system. Then, according to a proper hydrodynamic law (Darcy type applied to a cylindrical geometry), the velocity can be related to the pressure gradient by a straightforward correlation such as:

\[ v = -K^* \frac{dp}{dr}. \]  

(1)

where \( K^* \) is a proportionality constant which includes the actual permeability \( K \), as shown in the equation below:

\[ K^* = \frac{K}{\mu}. \]  

(2)

A typical porosity profile along the package radius is given in Figure 6, which was obtained by unwinding and weighing sequential portions of a bobbin made of acrylic yarn.
Figure 6: Porosity along Radius in a Yarn Package

Though the specimen cored out from a bobbin was subjected to a certain relaxation once paraffin was removed (while the yarn structure and crossing were perfectly maintained), its porosity was set by regulating the axial compression in the pressure drop apparatus given in Figure 2. By operating at porosity values in the range typical of packages the permeability $K$ was calculated by measuring the fluid pressure drop and by integrating Equation (1) along the test apparatus axis:

$$ p - p_{ext} = \left( \frac{\mu \dot{Q}}{K \pi R^2} + \rho g \right) (H - z) $$

where $p_{ext}$ is the pressure value downstream of the cylindrical fibrous element, $\dot{Q}$ is the volumetric flow rate, $H$ and $R$ are the height and the radius of the specimen, $\mu$ and $\rho$ are the viscosity and the density of the liquid, respectively; $\rho g$ is the term which accounts for the hydrostatic pressure in the system. The permeability profiles along the package radius at different temperature are shown in figures 7 and 8.

Figure 7: Permeability Profile VS Bobbin Radius at 27°C

Pressure Drop Measurements

The experimental tests on the pilot plant equipped with the capillary probe array permitted to measure the pressure drop values along the package radius at different axial positions. The pressure profiles were obtained by operating in a flow rate range corresponding to production operating conditions covering a temperature range from 25 to 95 °C. Figure 9 reports the mean pressure profiles along the radius package at 75°C by operating with an I/E liquor circulation. The mean value was averaged over different axial measurements, while the bottom position profile was not considered since a clear indication of liquid bypass at the interface with the lower package was detected according to the same measurements.

It is worthwhile noting that, at these conditions, the acrylic fibre approaches its glass transition temperature; indeed, at temperatures exceeding $T_g$, pressure drop profiles did not show regular trends because of some structural modification in the yarn.

Figure 8: Permeability Profile VS Bobbin Radius at 75°C

Figure 9: Mean Pressure Profile VS Radius at 75°C from 20 to 70 L/min; I/E circulation

Figure 10 shows pressure drop profiles obtained with an E/I liquor circulation. These measurements were carried out by reversing the flow in the same yarn package used in the previous step. A significant compaction was produced by the liquid flow, which could reach a maximum value of 68 L/min (corresponding to about 34 L/min/kg), while in the I/E circulation it was on purpose limited to 70 L/min not to modify the yarn structure. At these conditions the internal pressure drop distribution revealed to be substantially the same in both circulation directions.
Hydrodynamic Modelling

Since package permeability became available thanks to the two experimental methodologies, a hydrodynamic model was obtained by integrating Equation (1) and by making use of permeability versus radial position data. According to cylindrical coordinates the following pressure drop prediction was obtained:

\[ p - p_{\text{ext}} = \frac{Q}{2\pi l} \frac{\mu}{K(r)} \ln \left( \frac{R_2}{r} \right) \]  

(4)

Were \( R_2 \) is the bobbin external radius and \( K(r) \) is the calculated permeability profile.

Figure 11: Comparison between the Calculated and the Experimental Pressure Drop Profiles at 75°C, 60 L/min, I/E circulation

Figure 12: Comparison between the Calculated and the Experimental Pressure Drop Profiles at 75°C and 60 L/min E/I circulation

Ramp Temperature Test

The temperature profiles depicted in Figure 13 were obtained by imposing an in-to-out flow rate of 15 L/min/kg and a temperature gradient in the liquor of 0.8 °C/min.

Figure 13: Radial Temperature Profiles: Curves obtained at 60 s Time Intervals

The profiles with the experimentation time as a parameter revealed a 0.2 to 0.3 °C difference of temperature by moving from the internal to the external bobbin layers (thermocouples TC 2 to TC 7 progressively positioned from 94 to 99 mm radial distance from the former).
This temperature gradients may have a noticeable effect, if we consider that a 1°C difference may cause a dyeing rate increase of about 30%, due to variation of dye diffusivity in the fibre and alteration of the actual dyeing reaction rate. In each profile TC1 measured the liquor temperature in the spindle and TC8 measured the temperature in the liquid outside the bobbin. The profiles obtained by the thermocouple arrays positioned at different axial coordinates and along other radii are fully consistent with the results presented in Figure 13.

**Step Temperature Test**

Step tests were run with a sudden increase of temperature from steady state initial conditions to 60 °C. In order to accomplish each run hot water was abruptly fed from the heated tank to the kier by contemporarily discharge the liquid circulated in the apparatus: i.e. the circulation of an external liquid replaced the internal flow actuated by the pumping system. The test was performed by operating with a synchronic commuting of several valves which appear in Figure 4.

![Figure 14: Temperature Profiles during a Step Test; temperature step at 60°C; 20 L/min/bobbin; I/E circulation](image)

This kind of experimental method demonstrated occasional onset of fluid by-pass at the interface between two packages (see Figure 5), since in several in-to-out circulation tests TC 8 profile may even anticipate TC 6 and TC 7 profiles in the time scale (differently than in Figure 15).

If these hydrodynamic features are considered for a E/I flow in the apparatus, an unsteady mixing at the outside of a bobbin spindle pile may affect fibre coloration in the critical temperature interval. This consideration, together with the experimental E/I temperature profiles shown in Figure 16, should be taken into account at the equipment design stage. The very irregular temperature profiles might be caused by structure distortion in the bobbin (Duckworth, 1983).

A hydrodynamic model description of a package dyeing equipment can follow the schematization given in Figure 17: the connection piping between the heated tank (or in a standard equipment, the circulation pumping system) and the autoclave is definitely characterized by an thorough mixing, the hold-up volume given by the yarn packages act as a plug-flow system, while the empty space outside the package shows again a well-mixed behaviour.

![Figure 15: Mixing End-Effects during a Step Test; temperature step at 60°C; 20 L/min/bobbin; I/E circulation](image)
The temperature evolution along space and time was monitored by positioning capillary thermocouples at specific radial positions during bobbin winding. The dyeing pilot equipment was connected to an ancillary heated tank to feed hot water to the kier according to a stimulus (step or ramp) methodology. Subsequently, a data acquisition device recorded the system response by operating both with I/E and E/I circulated liquids.

The liquor circulation in the E/I flow appeared less neat than in the opposite case due to structure alterations in the package caused both by hydrodynamical compression and fibre thermal softening. Plug-flow was demonstrated to onset in the fibre package, while a well-mixed hydrodynamics prevailed upstream and downstream of the textile material, i.e. affecting the heat transfer in a subsequent dye bath-to-fibre interaction with liquor cycling. Occasionally, fluid by-pass at the interface between adjacent bobbins can be critical to dyeing quality.

REFERENCES


CONCLUSIONS

Unfortunately the partial bypass in the textile material (occasionally inferred from the temperature profile analysis) is not predictable in terms of entity and its occurrence should be reduced to a large extent both thanks to bobbin preparation or to accuracy paid to spindle assembly.
SIMULATION OF THE UNWINDING PROCESS

Yordan Kyosev*
Sven Hoffmann
Marion Strauf Amabile
Ingo Reinebach
Thomas Gries

Institute for Textile Technology, RWTH Aachen University
Elisbornsteinstr. 18, D-52062, Aachen, Germany
e-mail: {Yordan.Kyosev, Sven.Hoffman, Marion.Strauf, Ingo.Reinebach, Thomas.Gries} @ita.rwth-aachen.de

KEYWORDS
Balloon, non-stationary motion, unwinding, simulation.

ABSTRACT
A simulation of the unwinding process from wound packages is presented. The simulation differs from most other yarn balloon simulations which do not model the complete yarn path. The complete yarn path includes the wound package and the balloon. In the simulation described in this paper, the unwinding process and balloon building are presented as one dynamic process. The paper gives a short review of recent works in the area. The used mathematical model as well as some numerical difficulties are discussed. This work concentrates on the yarn motion during unwinding. The air resistance force and the friction between the yarn and yarn package are not included. It can be used for simulation and optimization of the unwinding process of high-speed unwinding machines and the weft insertion during weaving.

INTRODUCTION
The modeling of the yarn balloon is the subject of a large number of mathematical and mechanical investigations. The stability of the balloon, namely the equation of its stable motion are investigated in (Lisini, 2001), (Ghosh et al., 2001), (Batra et al., 1989a), (Batra et al., 1989b), (Fraser, 1992), (Fraser, 1996), the complete series of (Kothari and Leaf, 1979b), (Kothari and Leaf, 1979a) etc., and also by several other authors, using the equation of motion of one infinite yarn segment. All these works use separate equations for the separate zones of the yarn balloon. Because of this separation, the unique phenomena in the areas can be investigated in depth, but the model can not be applied for wider applications with more or different zones. In the last years (Przybyl, 1995), (Przybyl, 1998) etc. presented several works using mass-spring-systems as representations of the yarn, (Berger, 1996) discussed the problems in using such systems. These models can be applied in a wide range of industrial applications with complex geometrical conditions. In (Kyosev and Todorov, 2006) a mathematical model of the unwinding process where the yarn is modeled as a mass-spring system is presented. The equations of motion of the complete yarn length are integrated, including the yarn part on the bobbin, the free yarn length in the balloon and the yarn part behind the guide. This model successfully simulates the stationary yarn balloon as well as the switching between the single-loop and multiple-loop balloons. We extend here this model, implementing additional geometrical constraint for modeling of the yarn guide element. Some experimental results for the yarn tension are presented and the problems by their comparison with the simulated one are discussed.

PHYSICAL MODEL
The yarn is presented as set of particles, connected with line segments, as depicted on Fig. 2.b. The mass of the segments is transferred to the particles. All internal and external forces over yarn are presented as concentrated forces in these particles. The segments represent the presence of the tension forces between particles.

Assumptions
Following assumptions are done:

- The yarn is currently supposed to be of constant linear density. This means that all the particles have equal mass.

- Bending and torsion in the yarn is neglected. During the unwinding the yarn is normally in the tensioned state, where no significant bending and torsion effects are present (like in knitting loops). The neglect of bending and torsion also means that the
Figure 1: Zones with different boundary and initial conditions - A- the yarn guide element (eye), the balloon and the bobbin

- model is limited when dealing with highly twisted yarns.
- Air-resistance is currently neglected.
- Friction between yarn and wound package during sliding is neglected and will be subject of a separate work.

MATHEMATICAL MODEL

Equations of motion

The Newton law for force equilibrium for each particle is used for the description of the equation of motion of this element. The fixed Cartesian coordinate system is used. In matrix form, for the $x$, $y$, and $z$ coordinates the particle equilibrium is:

$$\mathbf{m}_i \cdot \ddot{x}_i = \mathbf{F}_i,$$  \hspace{1cm} (1)

where $\mathbf{m}_i$ is the mass of the current particle, $\ddot{x}_i$ is the acceleration of the particle and $\mathbf{F}_i$ is the resulting force at the current particle. This resulting force is calculated in the current case from the forces acting inside the connected segments - see Fig. 2 a. As mentioned in the assumptions, the friction forces during sliding on the wound packages is not taken into account. Since the bending effects are neglected and the segments have no mass, we do not need to use the Euler equation for the moment equilibrium.

Particle Equilibrium

The equation of motion for each particle is based on the force equilibrium. The resulting force $\mathbf{F}_i$ at particle $i$ is calculated as a vector sum of the forces from both neighboring segments $\mathbf{F}_{i,i-1}$ and $\mathbf{F}_{i,i+1}$ and the external force for the particle $\mathbf{P}_i$:

$$\mathbf{F}_i = \mathbf{F}_{i,i-1} + \mathbf{F}_{i,i+1} + \mathbf{P}_i,$$  \hspace{1cm} (2)

At a later stage all external forces will contribute to the resulting force. Currently only the gravity is taken into account.

Material Model

The forces during unwinding were measured. The results of the measurements show that the forces generally do not leave the linear area of the yarns’ load-deformation curves. For this reason, we implemented linear elastic behavior. If the forces are higher, a nonlinear material model can be implemented. The force in the line segments is calculated using the distance between particles. The initial unstressed segment between $P_i$ and $P_{i+1}$ of the unloaded yarn has the length:

$$L_0 = |\mathbf{P}_i \mathbf{P}_{i+1}|$$  \hspace{1cm} (3)

Deformed up to the current length $L$ under assumption...
of a linear elastic yarn, the tension force in the segment is:

\[
F_{i, i+1} = \begin{cases} \frac{E A (L - L_0)}{L_0} \frac{P_{i} P_{i+1}}{|P_{i} P_{i+1}|} & \text{if } L \geq L_0 \\ 0 & \text{if } L < L_0 \end{cases}
\]  

(4)

where \( EA \) is the spring constant, representing the common effect of the yarn elasticity module \( E \) and cross section area \( A \); \( P_{i} P_{i+1} \) is the vector with the starting point \( P_{i} \) and the end point \( P_{i+1} \). The pressure forces are not allowed as presented in equation (4).

**Boundary and Initial Conditions**

There are three regions with different boundary and initial conditions - the yarn on the bobbin, the free ballooning yarn and the yarn after the yarn guide. We start the simulation with zero initial velocities for all points, except the first particle, that is in the eye (yarn guide element). This particle has the initial velocity

\[
\dot{x}_{\text{start}} = [0 \ 0 \ V]
\]  

(5)

All particles that arrive at this point are allocated the velocity according to (5) as they pass the yarn guide. The initial coordinates of the balloon points are calculated using a quadratic parabola as initial approximation of the free motion yarn form. The balloon width, height and yarn end coordinates are used when fitting the curve for initial balloon geometry. The yarn geometry on the bobbin is calculated using the yarn guide motion low and the parameters of the bobbin. A short explanation of the used model for a few windings can be found in Kyosev et al. (2006).

**Motion constraints - yarn-guide-element**

One problem during the practical implementation of the current model was the modeling of the yarn-guide-element, which is an additional constraint for the yarn motion:

\[
\begin{align*}
&\text{if } H \leq z_i \leq H + L_{\text{guide eye}} \\
&\text{then } x_i^2 + y_i^2 \leq r_{\text{guide eye}}^2,
\end{align*}
\]

(6)

where \( H \) is the \( z \)-coordinate of the intake of the yarn guide eye, \( L_{\text{guide eye}} \) is the length of the eye element and \( r_{\text{guide eye}} \) is the radius of its hole. The obvious way here would be to model the contact between the yarn and the surface of the guide eye, but we used an elementary algorithm. It checks the condition (6) during the integration and if some particle is located outside the guide eye, its \( x \) and \( y \) coordinates are corrected so that this particle lies inside the eye. Since this correction is applied only once during several integration steps, for only one particle, the error from this intervention is negligible.

**Solver and Stability conditions**

We use explicit methods for the integration of the differential equation, which are conditional stable. The critical step size for the model was calculated according to (Halquist, 1998) for mass-spring systems:

\[
\Delta t_{\text{max}} \leq \sqrt{\frac{M}{K}}
\]

(7)

with \( M \) as the nodal mass and \( K \) as the spring constant \( K \) of the segments. Additionally, to ensure numerical stability we rewrote all the equations in dimensionless form, described in more details in Kyosev and Todorov (2006). We tested both Leap-frog and Verlet algorithms for solution of the differential equation (1) together with equations (2), (4). The difference in the calculated force between two algorithms is within 0.5%, when integration time step is \( \Delta t = 0.1 \cdot \Delta t_{\text{max}} \).

![yarn configuration](image)

Figure 3: Simulated yarn configuration for three time steps.

**RESULTS AND VERIFICATION**

The calculated configurations of the balloon for three time steps are presented in Fig.3, for a wool yarn, 120 tex, with unwinding speed 1244 m/min. The calculated
tension in five segments is presented in Fig. 4. The calculated force is zero when the segment is still over the bobbin, after that the force increases rapidly, when the particle is accelerated from the moving yarn part. After some oscillations the tension in each segment converges to one mean value which depends on the form of the yarn balloon. In Fig. 5 are presented yarn tensions for one segment, smoothed with mean filter, for three different unwinding velocities. The force shows that the initial increase of the force is higher for higher velocities and the particles need more time to get into quasi-stationary motion.

An experimental set-up for measuring the force during the unwinding is shown in Fig. 7 and consists of the frame, the bobbin (1), the yarn-guide-elements (4), the tension sensor (5). The laser distance sensor (2) (Baumer Electric) is used during longer unwinding tests to determine the current unwinding radius of the bobbin. The yarn is unwound with a constant velocity using the special unwinding machine "Enka Technika". Both sensors are connected by a PC with a National Instruments Data Acquisition Card and LabView Software. The acquired tensions are presented in Fig. 6 for the same velocities as the ones used for the simulation. The tension shows a certain periodicity whose period is equal to the unwinding time of one winding.

Currently it is not possible to compare the measured and simulated tensions, because the measured are taken from one fixed position in the space, where the segments are going through, and the simulation presents the force in each moving segment.

![Figure 4: Calculated force in the segment Nr. 20 during the time, for three unwinding velocities, Wool yarn 120tex, elasticity constant EA=0.55N/\%

![Figure 5: Calculated yarn tension for the yarn segments Nr. 1, 21, 41, 61 and 81 during the unwinding of the first two windings]

![Figure 6: Measured yarn tension at the yarn guide eye, for three unwinding velocities, after smoothing with mean filter. Wool yarn 120tex, elasticity constant EA=0.55N/\%]

**DISCUSSION AND NEXT STEPS**

The transient process from starting the yarn motion with zero initial velocities takes long time. At the same time the tension is at first higher than the measured one. To avoid this, two solutions are possible: a) a smoother start of the simulation, using increasing linear velocity or b) starting with non-zero initial velocities of the yarns. Both of these approaches require additional time where a) is more simple for realization, but takes a longer simulation time. Approach b) requires an additional calculation of the exact position and the velocities of the yarn particles during stationary motion. The
algorithm for keeping the yarn going through the yarn-guide-element is the next step for improvements. The current approach, based on watching the z coordinate, works well. In the case of more yarn guides with more general positions in space it will be more complicated. Implementation of the friction between yarn and package, as well as the air resistance are also needed, in order to model the process completely. In that way the model will be able to simulate the weft insertion process of air jet weaving machines, unwinding machines, as well as the motion of single fibers in the air flow.

CONCLUSIONS

The current work presents first results of the application of a new numerical model for yarn motion simulation. The yarn is modeled as a system of particles, connected by special only tension-able spring elements. A yarn-guide-element (eye) is implemented using an additional constraint equation. Numerical and experimental results are presented. A discussion of some problems and next steps are given.

ACKNOWLEDGEMENT

The Authors thank the Alexander von Humboldt Foundation for supporting the research stay of Dr. Y. Kyosev in RWTH Aachen.

REFERENCES


SIMULATION OF COMPLEX STRUCTURE YARNS

Fenglei Zhou
Department of Textile Engineering
University of Qingdao
266071, Qingdao,
P.R. China
E-mail: zfI200122@hotmail.com

KEYWORDS
Fancy Yarn, Mathematical Model, Space Helix, Matlab6.5, Computer Simulation.

ABSTRACT

A study is reported for mathematical model and simulation of complex structure fancy yarns. The investigated complex structure fancy yarns have a multithread structure composed of four components – one core, two effect, and one binder yarns.

In current research the precondition was accepted that the cross-sections of the both two yarns of the effect intermediate product in the complex structure fancy yarn remain the circles shaped, and this shape does not change while manufacturing the fancy yarn. Mathematical model of complex structure fancy yarn is established based on parameter equation of space helix line and computer simulation is further carried out using the computational mathematical tool Matlab6.5. Theoretical structure of fancy yarn is compared with an experimental sample. The simulation system would help for further the set of information in designing of new assortment of the complex structure fancy yarns and prediction of visual effects of fancy yarns in end-use fabrics.

INTRODUCTION

A number of theoretical and analytical studies have been reported in the literature devoted to the understanding of mathematical structures and simulation of fancy yarns. For a book (Gong and Wright 2002), the definition, structures and formation of fancy yarns were introduced in detail. Fancy yarns are those in which some deliberate decorative discontinuity or interruption is introduced, of either colour or form, or of both colour and form. The use of fancy yarns is solely for aesthetic effects. The first such yarns, such as dubbed, boucle, and knop types, became established textile novelties in the last century, but it was only in the 1970s that nontraditional yarns experienced a boom in the market (Testore et al. 1988). Woven fabrics and knitted materials using fancy yarns find applications in normal and high-fashion clothing, curtains, carpets, upholstery, wallpaper manufacture and many more.

A study was made (Testore et al. 1988) of the fundamental parameters involved in the production of certain fancy yarns. It was shown how use of these parameters enables an existing fancy yarn to be reproduced without resorting to empirical methods and new yarns exhibiting special pre-selected effects to be created. A geometrical model of complex structure fancy yarns was established (Petrunyté 2003) based on the precondition that the cross-sections of the all three yarns of the effect intermediate product in the complex structure fancy yarn remain the circles shaped, and thus shape does not change while manufacturing the fancy yarn. Based on the geometrical model, the dependencies were established of the structural and mechanical of fancy yarns upon the parameters of the manufacturing process – the delivery speed of fancy yarn, the rotational speed of hollow spindle, the speed of supply of effect component, which made it possible to design new fancy yarns with optimum properties for fabrics and knitted materials of clothing and decorative textiles (Ragašienė et al. 2003). The model was improved by introducing a new modeling method according to its process characteristics (Zhuge et al. 2004).

Forecasting of the properties of complex structure yarns and designing them, it’s very significant to choose properly and to co-ordinate parameters, which influence
structural, geometrical and mechanical features. Analysis of the structural, geometrical, and mechanical properties of fancy yarns as well as usage possibilities are very actual questions of today. In such a case, computer simulation will give a more realistic solution. For example, a computer imitation system was developed (Deng et al. 1994) for fancy yarns, which includes three main functions: manufacture of yarns, control of yarn colours and management of yarn resources. The Bezier curve illumination model, illumination factor template, HLS color model were applied (Zhuge et al. 2005) to the complex structure fancy yarns and further presented a scheme for numerical simulation of fancy yarn on the basis of the parameters: shape, material texture and colour. Up to the present, no model is found in the literature that is established on the space helix equation.

We proposed to study characterisation and computer simulation of fancy yarns commonly used in their manufacture. First we defined the characterisation codes. Then we investigated the structure and established theoretical mathematical model of complex structure fancy yarns. Finally we carried a numerical simulation of fancy yarns using Matlab 6.5.

RESULTS AND DISCUSSIONS
Theoretical modelling

The main idea of interpretation of complex structure fancy yarn model was that the following preconditions were made: firstly, the cross-sections of the yarns of core and effect components in fancy yarn remain the circle shaped, and this shape does not change while manufacturing the fancy yarn; secondly, the binder component winds the effect intermediate product at the place of core and effect component contact with screw winding; thirdly, the contraction of the binder yarn is not considered because of its insignificance. The first precondition is meaning especially in case when the yarns of core and/or effect component are spun yarns or twisted multifilament yarns that get less deformed in cross direction to compare with not twisted multifilament yarns or textured ones.

As the theoretical background of this study the geometrical model of fancy yarns is used. The structure of this model is definite by the geometrical properties of the core, effect, and binder component of complex structure fancy yarns as well as by the technological parameters of the manufacture (see Fig. 1). The yarn structures have necessarily been simplified, in order to make it possible for the diagram to show the important elements of the structure.

![Figure 1: Common View of Gimp Fancy Yarn Structure](image)

The fibres and filaments incorporated in the complex structure fancy yarn can be of different or same materials, colours, length, and thickness. Clearly, the fancy yarn can be produced in a variety of combinations. But depending on the end application of the textile material and the method of it manufacturing, the fancy yarn is produced accordingly. The mostly spread case of complex structure fancy yarns is that effect intermediate product consists of one core and one effect yarn or two core and one effect yarns. This structure especially is used trying to get expressive and combine effects of fancy yarn: slubs, knots, loops, spirals, snarls, waves, etc.

The structure of fancy yarn kind, when effect and core yarn’s diameter is larger to compare with the binder yarn’s diameter is shown in Fig. 2. The coil length of the binder yarn $L_b$ is equal to hypotenuse of triangle, which one cathetus is equal to coil pitch of binder yarn $h$, and the second one is equal to the length of projection of involute $p$. The value $h$ can be expressed in manufacture parameters of the fancy yarn: rotational speed of hollow spindle $n_s$ and delivery speed of fancy yarn $v_d$. So, $L_b$ is calculated according to the equation:

$$L_b = \sqrt{p^2 + h^2}$$  \hspace{1cm} (1)
Where \( h = \frac{v_d}{n_s} \), the index \( p \) (showed with dotted line in Fig. 3) is possible to determine from the equation suggested (Petrulytė, 2003).

\[
p = d_2 + 2 \times \sqrt{d_h \times d_k} + \frac{2d_h + d_k}{360^\circ} - 2 \times \arctan \frac{d_k - d_h}{2 \times \sqrt{d_h \times d_k}} - \arctan \frac{d_k - d_h}{2 \times \sqrt{d_h \times d_k}} + \pi \times \frac{d_k + d_h}{360^\circ} - \arctan \frac{2 \times \sqrt{d_h \times d_k}}{d_h - d_k} - \arctan \frac{d_k - d_h}{d_h + d_k}
\]

(2)

Figure 2: Relationship of Core Yarn and Binder Yarn

Considering the above preconditions, an expression as a function of time \( t \) for complex structure fancy yarns based on the helix equation is obtained as follows:

\[
\phi(t) = \left( r \times \cos(t), \frac{p}{2 \times \pi} \times \sin(t), \frac{v_s}{n_s} \times t \right), t \in \mathbb{R}
\]

(3)

Where \( r \) is the helix radius and \( h \) is the pitch.

According to the principle of manufacturing ring and hollow spindle combined system, the complex fancy yarn investigated is produced in two stages. Three yarns of widely differing count are plied together, thick around thin, and then reverse bound. Reverse binding removes most of the twist inserted during the first process. It is this removal of twist that creates the wavy profiles, since it makes the effect yarns longer than the actual length of the completed yarn. Since a binder yarn is needed to ensure that the stability of the structure, an approximate helix is formed between the binder yarn and the effect intermediate product. The helix radius is theoretically dependent on the diameters of core and effect yarns and the helix pitch is subject to manufacturing parameters such as rotational speed of hollow spindle \( n_s \) and delivery speed of fancy yarn \( v_d \).

Specify the parameters of the helix equation of fancy yarns:

\[
r = \frac{p}{2 \times \pi}
\]

\[
h = \frac{v_d}{n_s}
\]

Equation (4) gives the mathematical model of complex fancy yarns.

\[
\phi(t) = \left( \frac{p}{2 \times \pi} \times \cos(t), \frac{p}{2 \times \pi} \times \sin(t), \frac{v_s}{n_s} \times t \right), t \in \mathbb{R}
\]

(4)

Where \( \left( \frac{p}{2 \times \pi} \times \cos(t), \frac{p}{2 \times \pi} \times \sin(t), \frac{v_s}{n_s} \times t \right) \) is the helix expression of fancy yarns, \( p \) is the length of projection of involute of effect yarn, which is related to the overfeeding rate, \( v_d \) is the delivery speed of fancy yarn and \( n_s \) is rotational speed of hollow spindle.

Figure 3: Two-dimensional Model Used for Cross Section analysis: \( d_1, d_2, d_3 \) - diameters of yarns effect intermediate product, \( d_1 > d_3 \)

**Realization of simulation**

Based on the theoretical mathematical model of complex fancy yarn above, the simulation of complex structure is developed based on the mathematical tool software - Matlab6.5. Fig.3 shows the fancy yarn simulation user interface operated under Windows XP operating system, which includes two areas: Parameters area and Operation.
Area.
In Parameters Area, parameters of components of fancy yarn comprising materials, colour, and density, as well as the manufacturing parameters are predetermined by the operator. In Operation Area, three operations can be performed in turn. Firstly before inputting the parameters, the whole simulation system is strongly suggested to initialize and then the feasibility of parameters will be checked up in contrast with the parameter database of fancy yarns. Finally if the parameters input are applicable, simulation effect graph of end fancy yarns will be available through the operation of simulation. The investigated gimp structure fancy is simulated in this system and the comparison of the theoretical and experimental effects showed is shown in Fig. 4.

CONCLUSIONS

The simulation system of fancy yarns is based on of time for the design process of 20 per cent, which will involve the costs for this phase.
The simulation system represents original tools for the design and optimization of shapes of fancy yarns.

REFERENCES

Journal of Textile Research 15(6), 268-270.

manufacturing practice and theoretical mode of space helix, which applied in textile fields is opening a new challenge for fancy yarn manufacturers and designers. The simulation system is fully compatible with Microsoft Windows platforms, so does not involve supplementary costs for the user. The advantage of the system is that data are obtained in real time, from the simulation, not pre-calculated values, so a more approximate effect can be obtained in this simulation system. The utilization of simulation system in fancy yarn design ensures a reduction
Figure 3: Fancy Yarn Simulation User Interface

Figure 4: Theoretical Simulation Effect and Real Sample Effect
VIRTUAL TEXTILES AND OPTIMIZATION
A PLUG-IN TO DEFINE ASSEMBLING AND SEWING PARAMETERS FOR VIRTUAL CLOTHING SIMULATION

Rossella Mana  
Umberto Cugini  
Dipartimento di Ingegneria Meccanica  
Politecnico di Milano  
Via La Masa, 34  
20156 - Milano, Italy  
E-mail: rossella.mana@polimi.it

Alberto Carubelli  
Caterina Rizzi  
Dipartimento di Ingegneria  
Università degli Studi di Bergamo  
Via Marconi, 5  
24344 - Dalmine (BG), Italy  
E-mail: crizzi@unibg.it

KEYWORDS
CAD, apparel design, cloth models, virtual prototyping, assembly process.

ABSTRACT
This paper presents a plug-in, named Cloth Assembler, implemented in the framework of the Italian PRIN Project (Research Project of National Interest) VI-CLOTH (Virtual CLOTHing). The base idea is to allow the designer to interactively define the necessary information to assemble 2D panels on a virtual mannequin and to generate the 3D physical model in its initial configuration, initial step for the garment simulation process. Starting from 2D single pieces, ClothAssember allows a user, as a virtual tailor, to specify assembly rules among cloth panels (e.g., cut lines, dart, and buttons), insert accessories, such as zips and hooks, and, finally specify finishings on single pieces and the presence of different textile multilayer, pockets, reinforcement lines, etc. The plug-in can be easily integrated with any commercial 2D CAD system and represents the connection element between 2D cloth world and 3D physics-based modelling and simulation systems. It has been validated with three real test-cases, a T-shirt, a denim skirt, and a pair of trousers.

INTRODUCTION
Italian textile industry holds a share equal to 30 percent of the whole European production and it is second only to the automotive industry. Italian fashion is not only the consequence of styling, but also the result of an effort of design and testing, which involves different and integrated phases of the transformation cycle. The request for more competitive products leads the companies to increase the satisfaction of their customers: this is achieved with products distinguishing themselves for quality and fashion, but also for best fitting and comfort. At present, there are various software systems for 2D clothing design, grading, and so on, allowing great saving of time and material (e.g., FKGroup, Gerber Technology, and Lectra Systemes); however, there is a strong interest on integrated systems for the whole process including the possibility for the designer to work directly in a 3D environment, to design new garments, also made-to-measure (e.g., Browzwear, OptiTex, and Digital Fashion), (Hauth et al. 2002, House et al. 2000, Volino et al. 2000, Volino et al. 2000, Wang et al. 2002). With such tools garments behaviour (style and comfort) can be virtually simulated and validated, taking into account the mechanical properties of fabric and different human body or mannequin configurations. In such a context, the Italian Project VI-CLOTH (Virtual CLOTHing), resulting from a collaboration among 4 Italian universities (Università di Bergamo, Politecnico di Milano, Università di Firenze and Brescia), cloth manufacturers and CAD-CAM vendors, intends to develop an integrated 3D CAD system for the apparel virtual prototyping for real manufacturing purposes. Precisely, VI-CLOTH aims at defining a new computer-assisted design workflow in the clothing industry integrating reverse engineering and virtual prototyping techniques and tools as follows (Cambiasi et al., 2004) (Fig. 1):

1. a portable body scanner for human body data acquisition;
2. a module for parametric mannequin modelling, with 3D mannequin models derived from the acquired human body data;
3. a 3D garment modeller-simulator to generate from 2D unassembled cloth pieces the assembled garment in a 3D pre-simulated configuration and to predict the garment behaviour when placed on a 3D virtual mannequin.

Figure 2 shows the architecture of the 3D garment modeller-simulator system. Two main environments can be distinguished: one devoted to garment geometric modelling and another one for 2D-3D physics-based modelling and simulation. The first one is used for: the design of the 2D pieces composing a garment, the definition of finishing and assembly information on/among pieces and the successive mapping of the 2D pieces onto a 3D virtual mannequin. It includes three main modules:

1. a 2D CAD system, for garment pattern definition and editing of all 2D pieces;
2. an assembly module, named ClothAssembler, for the design of features and finishing details on each piece and the definition of assembly/connection elements relating the various pieces among each other. It also comprises a geometric tessellation module, to locally guide the folding in the successive 2D→3D mapping process.
3. a 2D→3D mapping module, named ClothMapper, to place the garment in a 3D assembled configuration.
Figure 1: Functional architecture of the VI-CLOTH project

Figure 2: Garment modelling and simulation system

In the physics-based environment, the 2D-3D physics-based model of the garment is generated and the final garment shape is then simulated according to material properties and mannequin configuration. It includes:
1. a physics-based modelling module, based on garment discrete representation by particle grids according to the particle modeling method (House et al., 2000);
2. a 3D garment simulation module, based on a constrained Newtonian dynamics model.

In this paper we focus the attention on the plug-in ClothAssembler specifically developed to define necessary data for garment assembly and on its experimentation.

TAILORING FEATURES FOR GARMENT DESIGN

Real clothing appears in a large variety of shapes and textile material, as result of several design, manufacturing and finishing operations. Moreover, it can be characterized by several aesthetic and functional elements that define garment peculiar look and function (Fig. 3).

Figure 3: Aesthetic and functional details in manufactured apparel: pockets, seams, zips, and buttons

In order to obtain an accurate description of real shaped corresponding geometrical and topological aspects should be defined within a CAD environment.

Adopting a feature-based approach typical of mechanical CAD packages, we have introduced the concept of tailoring feature to define all the elements and details that characterize a real garment. They introduce geometric-topological information to single cloth panels or groups of them, as a ‘semantic’ rule that can be applied, deleted, or repeated on the various panels. Obviously, these features are context-dependent, as the semantics underlying each feature class is dependent on the clothing manufacturing domain. Two categories of features have been considered:

- single panel features: aesthetic/functional rules applied to single panels, causing a local modification in the original geometry or material of the panels, e.g., local treatment, reinforcements, pleats, cuts, etc.;
- assembly features: rules for connecting (originally unassembled) panels, or (originally separated) parts of the same panel (e.g., by means of seams, small rigid constraints for fixing pieces together, etc).
We single out the different roles among the various 2D pieces defined in the early 2D CAD phase. In fact, some of them define the 2D geometry for the main visible textiles (named main panels), while other pieces will correspond to hidden/internal textile layers to be placed below a main panel or any union of them. A man suit, for instance, has its main textiles (e.g., cotton, leather, etc.) exhibited at the exterior sides; yet, these textiles can be composed of several interior textile layers such as linings, stuffings, reinforcements that, though not visible, still contribute to the final style and look of the suit. Our criterion is then to define tailoring features for the main panels; then, the corresponding rules will affect the possible layers located in the region where each feature is imposed.

Figure 4 shows the taxonomy of tailoring features defined for garment manufacturing (Carubelli et al., 2005).

CLOTHASSEMBLER

ClothAssembler has been developed according to the taxonomy above-introduced. It allows a user to mange (Carubelli, 2004):

- location and type of additional textile layers stuck or sewn on main textiles;
- different types of assembly rules, such as unary and binary seams, or darts, to join two disconnected borders of the same panel or two different panels;
- other assembly rules, such as zips, to temporarily connect two disconnected panel borders;
- location and type of border or internal reinforcements lines;
- location and type of border cut lines or internal cut holes, to partially disconnect parts, leading thus to a local change in the topological genus of the original panels;
- location and type of pleating lines to impose a fold on textiles;
- different types of point constraints, e.g., buttons and hooks, imposing a point-wise connection between two (border or internal) points belonging to the same panel or two different panels;
- different types of internal and external pockets, tightly-fitting the main textiles, or large.

Figure 5 shows a snapshot of the graphical interface of plug-in Cloth Assembler.

The main system functionalities are:

- file and edit functions, e.g., open, save, DXF-AAMA file import, XML file export, data and image printing options, and typical editing tools, e.g. cut/copy/paste and undo/redo;
- 2D view modes, e.g. global layout and detailed views, zoom/pan/translate/rotate and show/hide functions;
- 2d pattern menu, to define some properties of the single piece such as the type of fabric or the texture. It is also possible to modify reference points on the selected piece, e.g. notch on the edge and holes;
- garment making modality, allowing the insertion of all tailoring features through the selection of each panel or between 2D pattern pieces, e.g. layers, seams, darts, zips, point constraints;
- 3D Modelling menu, for the panel location in 3D, based on the assignment of Cartesian/anthropometric reference data.

TESTING PHASE

ClothAssembler has been validated by testing its functionalities on garment models of real interest and production. We made reference to six test-cases, three female cases and three male ones, defined to test the garment modelling and simulation system. The considered test cases have been classified according to different types of complexity depending on the geometry of each panel, sewing/assembly rules and the clothes’ aesthetical/functional details. Table 1 shows, for each test-case, the geometric and construction elements that have been considered.

<table>
<thead>
<tr>
<th>WOMAN</th>
<th>SHAPE AND STRUCTURE</th>
<th>PROCESS</th>
</tr>
</thead>
<tbody>
<tr>
<td>SKIRT</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>PANTS</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>JACKET</td>
<td>X</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>MAN</th>
<th>SHAPE AND STRUCTURE</th>
<th>PROCESS</th>
</tr>
</thead>
<tbody>
<tr>
<td>T-SHIRT</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>PANTS</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>JACKET</td>
<td>X</td>
<td>X</td>
</tr>
</tbody>
</table>

Table 1: Test cases
For the preliminary validation of the ClothAssembler in garment design and assembly, we have selected three test-cases: a T-Shirt made by cotton (Figure 6-A), a denim skirt (Figure 6-B), and trousers (Figure 6-C), starting from the definition of their 2D pieces using a 2D CAD system for the clothing sector.

In the following section we describe in details the 2D design and assembly stages performed for the testing.

Figure 6: Test-cases: (A) T-Shirt, (B) Denim skirt and (C) a pair of Trousers

Design and development of 2D pattern panels
The geometry of the 2D pieces composing a garment is defined using a commercial 2D CAD system, namely TexWinCAD by the Italian Company F.K. Group. The methodology used for the digital models definition is derived from the traditional paper patterns typically used in the sartorial sector. Thanks to this CAD system, the different 2D panels have been designed without seams, prerequisite for the successive 3D virtual simulation. According to this operation, an external sewing border has been added along the whole perimeter of the silhouette. Finally, the panels have been saved in DXF AAMA format, a neutral file format widely used for data exchange in the clothing sector.

Figure 7, 8, and 9 show the resulting 2D patterns of the T-Shirt, the denim skirt and the pair of trousers.

Figure 7: Design of 2D T-Shirt pattern

Definition of assembling and sewing parameters
In the previous design phase the main focus is the definition of geometry of each garment pattern while the connection between these pieces is not considered. In this step, Cloth Assembler allows us to define the relationships between the single panels, and to insert all those accessories whose presence had been planned in the design phase by an accurate adjustment of the geometry of the panel.

The T-Shirt has a relatively low level of design and manufacturing complexity; in fact, it is composed of four pieces: two front and back pieces, and two sleeves.

Figure 8: Design of 2D denim skirt patterns

Figure 9: Design of 2D trousers patterns

This model is characterized by ten seams, four of which connecting the front and the back pieces, and six connecting the sleeves to the main body (Fig. 10).

Figure 10: T-Shirt tailoring features
Seams between the front and back pieces were quite easy and intuitive to obtain, while seams between the sleeves and the main body are, on the contrary, more difficult as they connect borders with different lengths (in fact, the sleeve border is longer than the arm pit-girth): as a tailor manually fix reference notches along the borders to proportionally distribute the looseness of the seam, the same was done – and easily – by ClothAssembler with automatic notch insertion. Moreover, this garment does not present any constraints, e.g. buttons nor zip, and particular attention during the manufacturing and finishing processes.

The denim skirt is a more complex test case. It comprises twelve pieces, including the waistband. This model is characterized by fourteen seams, four pockets and two constraints, button and a partible zip (Fig. 11). For example, the lower horizontal side of the waistband, for instance, is sewn with the upper side; then, the entire length is sewn with four skirt’s pieces. Pockets require an evaluation of the internal polyline onto which they are sewn. ClothAssembler facilitates the pocket localization against the polyline by using reference drill holes (blue, in Fig. 12) along the pocket opening. Figure 12 shows the detail of some tailoring features.

The last test-case, a pair of trousers, is composed of seven pieces, including the waistband. This model is characterized by twelve seams and a partible zip. (Fig. 13)

The assembling operation of the three test-cases required only few minutes, i.e. about 3’-4’ for the T-Shirt, 5’-6’ for the pair of trousers and 8’-9’ for the denim skirt. This is due to the easiness of use of the Cloth Assembler module which requires just a basic knowledge of traditional garment manufacturing.

ClothAssembler includes also a geometrical tessellation module to generate 2D panel’s triangulation necessary to drive the 2D→3D mapping phase onto the virtual mannequin. The module for 2D tessellation of the pattern pieces includes the following functionalities:

- Manual tessellation, for a free definition of non-overlapping facets and triangles within the panels by interactive user-defined point insertion and vertex connection.
- Automatic tessellation, a fast procedure based on prearranged algorithms, such as unstructured Delaunay triangulation or structured mesh generation
- Semi-automatic tessellation, a mixed generation approach with previous subdivisions in main regions, each associated to local manual or automatic tessellation.
- Modification of tessellation, for local correction of an already existing/created tessellation, based on fully user-controlled or automatic procedures.

In all cases, 2D tessellation is constrained, since edge, and curve points, notch and drill-hole points introduced in the previous design and assembly phases are required to be nodes for some triangle/facet. Additional constraints can be imposed to better control the shape, location and orientation of facets.

Figure 13 shows all steps from 2D patterns design to 3D simulation for a pair of trousers based on the integrated used of the modules composing the physics-based modelling and simulation environment (Fig. 2).
CONCLUSIONS

In this paper we have presented a plug-in named ClothAssembler, implemented in the framework of the Italian Project VI-CLOTH. The system allows a user to define assembling and sewing parameters in 3D of garment models of real interest and production from single pieces that compose it, to insert aesthetic/functional details such as zip, pockets, pleats, reinforcements lines, cuts, darts, constraints, the presence of different textile multi-layer and, finally to specify the properties a single panel, such as the type of fabric.

At the present, however, Cloth Assembler is not able to deal with all the possible assembling cases which might occur. For this reason, new tailoring features will be implemented, e.g. further seam and pocket types, which will make it possible using this plug-in also for defining the assembling and sewing parameters of more laboured test cases, e.g. jackets and coats. The system has been integrated in a 3D environment for the virtual prototyping of apparel for real manufacturing purposes that includes physics-based modelling and simulation of assembled garments placed in a 3D configuration to verify the final simulation effect that all virtual prototyping process of garment.

ACKNOWLEDGEMENTS

This work has been carried out in the framework of the Italian PRIN Project VI-CLOTH, funded by the Italian Ministry for University and Research. The authors would like to thank the F.K. Group, F.Lli Corneliani and Marianna Confezioni, and all colleagues from Università di Bergamo, Politecnico di Milano, Università di Firenze and Università di Brescia, Italy. A special thank to Marzia Fontana for her invaluable contribution to the project.

REFERENCES


BIOGRAPHY

ROSSELLA MANA is a Ph.D. candidate at the Department of Mechanical Engineering, Politecnico di Milano, Italy. She received her M.Sc. Degree in Industrial Design and Master in Virtual Prototypes in Industrial Design at the same university. She is author and co-author of several publications in the area of industrial design, modelling and simulation.

ALBERTO CARUBELLI received the M.Sc. Degree in Computer Science Engineering from University of Parma. He has competences in computer graphics and geometric modelling.

CATERINA RIZZI is full professor at the Faculty of Engineering, Università degli Studi di Bergamo, Italy. She participated in several national and international projects in the area of modelling, simulation and CAD applications. She is author and co-author of more than 120 publications.

UMBERTO CUGINI is full professor at the Department of Mechanical Engineering, Politecnico di Milano, Italy. He participated in several national and international projects in the area of modelling and simulation, and coordinated several national projects in the area of product design, engineering and robotics. He is author and co-author of more than 150 publications.
A VIRTUAL PROTOTYPING TOOL FOR KNITWEAR SIMULATION

Sylvia Chalençon, Olivier Nocent, Jean-Michel Nourrit and Yannick Remion
Department of Computer Science
Reims Institute of Technology
rue des crayères BP 1035
51687 Reims cedex 02
France
E-mail: {sylvia.chalencon | olivier.nocent | jm.nourrit | yannick.remion}@univ-reims.fr

KEYWORDS
Knitwear simulation, virtual prototyping, continuous modelling.

ABSTRACT
This paper both presents a 3D geometric model of knitted structures based on technical data usually used by knitted cloth experts and a physically-based model for knitwear simulation. This physically-based model relies on 3D curves modelled as successions of spline segments. It insures that at each time step the curve shape conforms to its spline definition, and thus that every property implied by the nature of the chosen spline model is verified. This fact is achieved by the animation of the spline control points. However, these control points are not considered as material points but as the degrees of freedom of the continuous object. The knitwear simulation suite encompasses topological and geometrical modelling of the rest state, mechanical modelling (including stitches bindings) suitable for our custom simulation engine based on Lagrange equations.

INTRODUCTION
Largely widespread in the main industrial sectors, the use of software for computer-aided design (CAD) remains marginal in the textile industry. These tools are rather seldom in the “stitch field” because of the complexity of the knitting process and the computing power hitherto insufficient. This project, in collaboration with the French Textile and Apparel Institute (I.F.T.H.), follows upon the two PhD theses of J.M. Nourrit and O. Nocent which introduced algorithms and models for knitted cloth simulation, both geometrically (yarn trajectory, stitch interweaving) and mechanically (elastic behaviour, back and forth extension). This work aims at developing a software prototype for static and dynamic simulation dedicated to knitted cloth designers. This software will allow 3D modelling of knitted structures based on technical data usually used by knitted cloth experts. However, 1st of January 2005 marked the end of the ‘multifibre agreement’ of 1975, which imposed quotas to emergent countries. Chinese textile exportations towards Europe then strongly increased in the first quarter of 2005. An article of Radio France International published on internet the 25th of April 2005 announced a raise of 534 % for pullovers exportations and 183% for socks exportations. These statistics show clearly that the European knitting sector for clothing is struck hard by this competition. Thus, the future of knitting in France could, according to our partners of the I.F.T.H., lie in technical knitting. The Textiles of Technical Use (TTU) are textile materials answering high technical and qualitative requirements (mechanical, thermal or electronic properties, durability and so on) conferring an aptitude to them to be adapted to a technical function and its environment. But the achievement of these requirements relies on an advanced technology based on a computational model of knitted structures.

CAD FRAMEWORK FOR KNITWEAR CONCEPTION

Because of the complexity of knitted clothes, there is any complete CAD solution adapted to knitted structures at this time. The most recent works in modelling, such those of Meißner and Eberhardt (Meißner and Eberhardt 1998) for example, remain limited to knitted structures with simple topologies, and without holding account of the mechanical characteristics of the yarn, nor of the fundamental parameters of knitting (such as gauge or length of yarn absorbed by a stitch). In addition, the generated models are not three-dimensional and are used for diagrammatic representations limited to error control performed by specialists.

Anatomy of a stitch
A stitch is subdivided into three parts: its head (upper part), legs (middle part) and feet (lower part) (Figure 1).

![Figure 1: Stitch Parts](image)

Since a piece of knitting is being made of intertwined stitches, one sees the face side of a stitch when its legs are visible whereas its head and feet are hidden (Figure 2a). Conversely, one sees the reverse side of a stitch when its legs are hidden whereas its head and feet are visible (Figure 2b).
The most basic piece of knitting is "jersey": the fabric is fully made of face stitches. As a consequence, one only sees face stitches whereas the other side exhibits only reverse stitches. The tuck stitches are obtained by doubling the head of the last manufactured stitch (Figure 3), thus producing the tying usually used for full cardigan rib or half cardigan rib.

Our geometrical model of knitted cloth relies on three-dimensional spline curves representing the trajectory of each yarn within the knitting. The 3D positions of control points of these curves are mainly defined by the tying type (way of knitting, such as for example jersey, rib, cable stitch) and by the geometrical and mechanical characteristics of used yarns. Considering the huge variety of tying and yarns employed by the stitch industry, an automatic modelling tool is necessary to avoid using static libraries of objects (too rigid and insufficient because of the large amount of tying/yarn combinations. In addition, the use of different tying within the same knitting is responsible of stitches deformations, which must be generated dynamically.

Starting from the work of Meißner et Eberhardt (Meißner and Eberhardt 1998), we carry out a 3D modelling of knitted structures application. The stitches of knitting consist of interlaced loops of yarn. According to tying considered, these loops can undergo more or less marked deformations. Initially, we build an non deformed loop, such as one can observe some in a jersey-based fabric, which is the basic tying of the knitting industry.
or rear needle bed). The contact points thus make it possible to define the stitches geometry.

The first stage completed, the BP are placed while holding account of the type of stitch and the neighbours or of the operation carried out (transfer to opposite needle bed or shift of a needle bed) but also of the gauge and spacing of needle beds.

The second stage consists in determining the trajectory of the wire within knitting; this operation mainly relies on the tying type (in the case of a face stitch, legs are on top whereas for a reverse stitch, the head and the feet are visible) constituting the fabric (the yarn radius is then supposed constant). We use cubic spline curves because of their simplicity of handling 3D trajectories modelling.

The principal purpose of our application is thus to reveal the knitting topology starting from yarn path notation drawn on our 3D modelling application (Figure 8).

Then we translate the drawing using an alphabet we have created. The yarn path notation (or the “vermiculation” language) is quite universal since it is mainly used by professionals of the knitting industry. This is another advantage compared to the software of Meißner et Eberhardt (even if they could undoubtedly have made this modification without too many difficulties) which received owners data. The various effects of knitting such as load, transfer and shifts of needle bed, deactivation of needles, can be generated. The alphabetical coding that we determined to translate the “vermiculation” language in a knitting definition text-based format is as follows:

<table>
<thead>
<tr>
<th>«vermiculation» symbol</th>
<th>Associated letter</th>
</tr>
</thead>
<tbody>
<tr>
<td>Front Side Stitch</td>
<td>G</td>
</tr>
<tr>
<td>Rear Side Stitch</td>
<td>E</td>
</tr>
<tr>
<td>Front Side Tuck</td>
<td>V</td>
</tr>
<tr>
<td>Rear Side Tuck</td>
<td>N</td>
</tr>
<tr>
<td>Transfer to Front Needle Bed</td>
<td>T</td>
</tr>
<tr>
<td>Transfer to Rear Needle Bed</td>
<td>T</td>
</tr>
<tr>
<td>Shift of Front Needle Bed</td>
<td>R</td>
</tr>
<tr>
<td>Shift of Front Needle Bed</td>
<td>R</td>
</tr>
<tr>
<td>Inactive Needle</td>
<td>X</td>
</tr>
</tbody>
</table>

Table 1: Our alphabet

Note: shifts of needle bed can be made on the right or on the left, that is determined by the number which follows the R or r (if the number is positive the shift is done of as many needles towards the left if not the right-hand side).

Our application provides a fully interactive framework for knitting modelling since it allows realtime 3D navigation to relocate the knitting and to turn around. In addition, if one changes the content of the text file, the digital knitting is updated immediately.

Figure 8: Our 3D Modelling Application.

Here is an example of a 3D digital 2x2 rib built from the sequence below:

Figure 9: 2x2 Rib

PHYSICALLY-BASED MODELLING FOR KNITWEAR SIMULATION

As cloth simulation is such an important matter, numerous solutions have been proposed. More precisely, numerous solutions have been proposed in the case of woven cloth (Terzopoulos and Fleischer 1988, Breen et al 1994, Eberhardt et al 1996, Volino and Thalmann 2000), but, not surprisingly, very few deal with the case of knitwear simulation. The fact that knitted cloth study is still marginal can easily be explained by the complexity of the knitted cloth structure of which the stitches fitting plays an important part in the global dynamic behaviour. This reflection got us to develop a model for knitwear at the mesoscopic scale, meaning fibre scale (the mesoscopic scale consists of an intermediary scale between macroscopic and microscopic scales). This choice seems to be justified and above all necessary. One could mention a similar work of Eberhardt (Meißner and Eberhardt 1998) which introduces a
topological knitwear modelling tool more than an actual knitwear simulation platform. A more recent paper of Chen et al. (Chen et al 2003) presents a knitwear model that relies on a skeleton surface made of masses and springs. Using appropriate energy functions, the model can be animated using the basic laws of dynamics. But, as the authors confess, this kind of modelling is not able to simulate the motion of cloth inferred by the yarns and their stitches. Our knitwear model is respectful of the intrinsic yarn properties (mass repartition and elasticity) and is the only one able to take into account the stitches structural complexity.

Since knitwear simulation must then rely on yarns and stitches modelling to ensure sufficient realism, yarn mechanical modelling must encompass deformable geometry, mass repartition along this geometry and internal strains. Then stitches modelling can be achieved with geometric constraints and/or external strains imposed on the yarns.

**Yarn modelling**

The main geometric feature of the yarn lies in its global shape defined by a three dimensional curve called its axis. Three dimensional parametric curves are important topics in Computer Graphics and Computer Aided Geometric Design (see (Watt 1989, Farin 1990) for example) as they can be used to model manufactured objects such as ropes, threads, thin bars and so on. Among these, curves called “splines” defined as parametric functional combinations of 3D control points encounter a real success in the Computer Graphics community for their modelling facilities. Their most interesting properties in the scope of yarn axis modelling are smoothness control, realistic shapes, and, most of all, limited number of control points. This finite and limited number of control points infers a related finite and limited number of degrees of freedom (DoF) for the axis shape. This finite number of DoFs is necessary in order to achieve a numerical animation.

At this stage yarns are curves whose shapes are both realistic and suitable to numerical animation. To achieve mechanical modelling, one has to model mass repartition over the shape and internal strains, leading to the “Dynamic Material Spline” entity (Remion et al 2000, 2001), also called DMS. We then recall the main features of our DMS model.

Numerous spline models have been proposed. This diversity of the existing models eggs us on rendering our DMS “generic” to encompass as many existing spline models as possible. In this scope, the diverse models of either family may be perceived as successions of one or more curves (which we call “spline segments”) defined as parametric functional combinations with a set of blending functions \( b_i(s) \) of a common set of 3D moving control points \( q_i(t) \).

The time dependent position of a point on such a curve depends on its segment number \( j \) and its parametric position on this segment \( s \in [0,1] \). The position \( p_j(s,t) \) is then given by the following equation:

\[
p_j(s,t) = \sum_{i=1}^{n} b_i(s) q_i(t)
\]

As previously noted, dynamic animation of a spline-like object should preserve its geometrical properties over time. One simple method then consists in animating the control points of the parametric curves.

This can be achieved with dynamic animation of material control points. But we believe that the concentration of the object mass on a discrete set of 3D points is unrealistic and even more so if one remembers that some of these control points could not lie on the object for approximating spline curves.

Hence, we propose another method somehow similar in its assumptions to the D-NURBS proposed by D. Terzopoulos and H. Qin in (Qin and Terzopoulos 1996). These assumptions consist in considering the control points as the degrees of freedom of the non-discrete (say “continuous”) object. As mass repartition remains continuous, these methods are more accurate than the more straightforward animation of a set of material control points.

In the particular case of spline like objects we consider that thickness is negligible and, so, model mass repartition as \( n_s \) mass density functions (one per spline segment):

\[
\mu_j : [0,1] \times \mathbb{R} \rightarrow \mathbb{R}
\]

\[
(s,t) \mapsto \mu_j(s,t)
\]

As spline segment lengths can evolve in time, this implies that mass density per unit length can evolve too, even if \( \mu_j(s,t) \) is actually invariant in time (this time invariance simply meaning that matter is not sliding along the curve and is tied to its parametric position \( s \)).

But in (Nocent and Remion 2001), we propose a more “user-friendly” definition of mass repartition based on a per unit length density \( \rho_j(s,t) \). These two density functions can be related according to the current DMS arc length \( l_j(s,t) \) defined as:

\[
l_j(s,t) = \int_0^s \lVert \frac{\partial p_j}{\partial s}(u,t) \rVert \, du
\]

where \( dl = \left\lVert \frac{\partial p_j}{\partial s}(s,t) \right\rVert \, ds \neq ds \).
Initially, we used non linear elongation springs between consecutive control points in order to simulate internal strain (Figure 11). Since (Nocent and Remion 2001), we are now able to deal with a continuous energy of deformation based on the theory of elasticity for finite displacements. This expression is fully compatible with our DMS model, much more realistic and also numerically efficient. The physically based model finally achieved; we now have to choose the dynamic equations most suited to its nature. Since it is continuous and non-rigid with a finite number of degrees of freedom (3n), analytic mechanics seem best suited than Euler or Newton/Euler formalisms dealing respectively with material points and solids. We therefore choose to use Lagrange equations:

\[
\frac{d}{dt} \frac{\partial K}{\partial \dot{q}^*_{\alpha}} - \frac{\partial K}{\partial q^*_{\alpha}} = Q^*_{\alpha} \quad \forall \ i \in [0,n], \alpha \in \{x,y,z\}
\]

where \( K \) is the kinetic energy function, \( q^*_{\alpha}(t) \) are the scalar degrees of freedom and \( Q^*_{\alpha} \) is the power rating of the external forces in the virtual movement instilled by \( q^*_{\alpha} \).

![Figure 11: Elongation Springs.](image)

In the peculiar case of DMS, Lagrange equations then yield to the following linear equations system of the unknowns \( \dot{q}_{\alpha}(t) \):

\[
M \ddot{q}_{\alpha} = Q^*_{\alpha} \quad \forall \alpha \in \{x,y,z\}
\]

where \( M_{\alpha} = \sum_{j=1}^{n} \rho_j(s)h^2\alpha(s)h^2\alpha(s)ds \)

The temporal integration of this system is a classical mathematical problem, for which a set of well-known solutions exists (Press et al 1992).

**Knitwear modelling**

Obviously the knitwear model conforms to the ideas developed in the introduction. Thus, a piece of knitwear is modelled as series of yarns. Each yarn is defined by a DMS featuring its axis and further data featuring threads properties. We choose to use Catmull-Rom splines that are cubic interpolating curves. We use an enhanced version of these curves where a tension parameter has been added to the equation (Vince 1992). In the case of yarn-based knitwear simulation, the question of initializing position or shape isn’t trivial. These “initial yarns” should be suitably intertwined to form the right stitches. In fact one has to build a topologically accurate and geometrically acceptable series of shapes, one for each yarn. Ideally these initial shapes should be mechanically refined to enhance their geometrical accuracy under the initial strains. In fact, this initially constrained state could be obtained after a few step of dynamic animation starting from repose. We thus chose, for the moment, to initialize in this repose state. The topologically accurate and geometrically acceptable initialization is obtained by rough observation of the basic knitwear feature: a jersey stitch.

If one examines the projection of a jersey piece of knitting on a plane parallel to the cloth, some interesting properties of the stitches appear, which can also be seen on Figure 12. Segments between points 3 and 5 are symmetrical from point 3 to the segments between points 1 and 3. These two points correspond to the bonding points generated by our CAD tool described above. Moreover, the stitch presents an axial symmetry.

![Figure 12: Stitch Symmetries.](image)

So, we only have to build the planar geometry for spline segments between points 1 and 3 to obtain the entire stitch planar geometry thanks to those symmetries. The control points positions are determined as a function of the yarn thickness and the stitch density.

As shown in Figure 12, a stitch is made of 8 Catmull-Rom spline segments controlled by 11 points. Such a stitch is only part of a yarn, and is adjacent to up two other stitches along its yarn. Obviously these adjacent stitches will share some of its control points, namely points numbered 0, 1, 2 for the preceding one and points 8, 9, 10 for the following one. In order to simplify, yarn warping from row to row is forgotten and each row of \( N \) stitches is modelled as a dedicated yarn composed by \( 8N \) spline segments for a grand total of \( 8N+3 \) control points.

Rows intertwining is then obtained by a convenient distribution of the elevations of control points along the third dimension (Figure 13).

At this stage we have modelled knitwear as series of yarns (one for each row) with convenient initial shapes and intrinsic mechanical properties. Our model still lacks the most significant feature of knitwear overall cohesion: the stitches bindings. These bindings are dealt with according to their implied yarns contacts.
Contacts simulation uses jointly:
- Holonomical constraints aiming at preserving the barycentrical coordinates of points $B_1$ and $B_2$ in, respectively, $(A_2B_2C_2)$ and $(A_1B_1C_1)$ (Figure 14).
- Bi-directional springs set as in Figure 14, preventing the yarns from interpenetrating each other, and thus preserving the yarns relative orientation.

We know that this way of modelling contact between yarns is not general enough especially in the case of stitches load. Remion introduced the concept of ‘free variable constraints’ which is an elegant way to formulate complex constraints equations. This theoretical framework was successfully used by Lenoir in (Lenoir et al 2004) to integrate sliding contact constraints in a surgery simulation. In a near future, we want to use this kind of constraints for stitches bindings.

**CONCLUSION**

In this paper, we first presented a CAD framework for knitwear conception based on an industrial standard. This tool allows creating any kind of knitwear pattern using a normalized technical language called the “vermiculation” language. A 3D visualization tool is also provided to see and modify the knitwear structure in real-time.

In the second part, we reminded the different steps of the creation of a physically-based model for knitwear from yarn modelling to stitches bindings. This mechanical model is already functional but only for a single knitwear pattern: the jersey pattern.

In a near future, we aim to merge these two independent modules in order to achieve a complete virtual prototyping tool relying on technical data and able to simulate the mechanical behaviour of virtual knitwear patterns.

**REFERENCES**


**AUTHOR BIOGRAPHY**

SYLVIA CHALENCON was born in Reims, France and went to the University of Reims, at their Troyes site, where she studied Computer Science and Artificial Intelligence and obtained her Master degree in 2004. She is now preparing a PhD thesis at CReSTIC-LERI (the Computer Science Laboratory of the University of Reims) in Computer Graphics and Simulation. Her research topics are related to knitwear simulation, continuous modelling and computer animation.
FABRIC BEHAVIOUR UNDER BI-AXIAL TENSILE TESTING
BI-MODEL IN FEEDBACK CONFIGURATION

Agnieszka Cichońska
Technical University of Lodz
Faculty of Textile Engineering and Marketing
Department of Clothing Science and Technology
Ul. Żwirki 36, 90-942 Łódź, Poland

Pascal Bruniaux
Vladan Končar
Laboratoire GEnie et Matériaux TExtiles
(GEMTEX), UFR EA2161
École Nationale des Arts et Industries Textiles (ENSEAIT)
9 rue de l’Ermitage, BP 30329, 59056 ROUBAIX Cedex 01, France

KEYWORDS
Bi-model, Bi-axial tension, Fabric simulation, Identification, Genetic algorithms

ABSTRACT
The aim of this article is to develop a model related to dynamic behaviour of the fabric, under strong constraint, in order to analyze the problems of surface deformation. In practice, we use a Bi-axial tensile testing machine to generate efforts in the plane of fabric. The theoretical study is based on setting up a model of representation, enabling the simulation of material deformations, in the weft and warp directions. After a short introduction describing the overall model of fabric under specific constraints, its detailed presentation is proposed. This global model is composed of 2 sub-models. The first one is dealing with the phenomena generated in warp threads and the second with the phenomena generated in weft threads. These two sub models are interconnected by the feedback effect coming from interactions of warp and weft threads during their contacts in the fabric structure. In the second part of the paper, strategy of model parameters identification is proposed. The relatively significant number of parameters requires the use of identification technique based on Genetic Algorithms. This method alone is not sufficient to identify the global model correctly. Also, it was necessary to adapt the identification algorithm following a specific strategy associating the concept of Bi-model.

INTRODUCTION
The fabric characterization can be carried out under various mechanical constraints. The requests which are regularly set up in the field of flexible materials are in general associated with mechanical tests such as: axial traction plane, Bi-axial traction, bending, torsion etc. The Bi-axial traction is the most important for us. The experimental part of this test is to draw on both sides of fabric, following its two principal axes (warp and weft), at constant speed until the rupture. Since the space constraints are in the same plane, it is possible to model the overall dynamic behaviour of fabric. The diagram in Figure 1 proposes a scheme of the global model which can be represented by a multi input multi output linear system. These inputs and outputs (E1, E2, S1 and S2) are interconnected between them by two feedback loops translating the internal phenomena with the structure of material (J.-Ch. Gille and al. 1971).

![Global model](image)

MODELLING OF BI-AXIAL BEHAVIOUR
The global fabric model is composed of two parallel sub-models. The first sub-model translates the deformation of warp without the weft thread. Similarly, the second translates the deformation of weft threads without the warp. The process of weaving generates threads intersections that imply the addition of two feedback loops (FB1, FB2). Sub-models were conceived in order to take into account the viscous elastic dynamic behaviour of threads. From the mechanical point of view, two representations are possible to model this type of behaviour. In the first case, the model is represented by putting in parallel an elastic element (spring and a viscous element) type of dashpot (model of Kelvin Voigt). In the second case, the model is realized by putting in cascade an elastic element and a viscous element (model of Maxwell). Separately, these models are assimilated to first order systems (Figure 2) such as:

\[ \sigma_1 = E_1 \varepsilon_1 + \frac{\eta_1}{\gamma} \frac{d\varepsilon_1}{dt} \]  \hspace{1cm} (1)

\[ \sigma_2 = E_2 \varepsilon_2 \delta^{(i)} \]  \hspace{1cm} (2)
The experimental shape of strain-stress diagrams shows that the best adapted model is a second-order one. This adapting is possible by putting in series the Voigt Kelvin and Maxwell models. Figure 2 represents our constraint model according to the deformation of fabric. The equivalent (J. Baillou and al. 2002) transfer function of this new system is given below:

\[ H(s) = \frac{\sigma(s)}{\varepsilon(s)} = \frac{(E_1 + \eta \varepsilon) \eta E_2}{E_1 E_2 + E_2 \eta s + E_1 \eta s + E_1 \eta s + \eta \eta s^2} \]  

or

\[ H(s) = \frac{(E_1 + \eta \varepsilon)K}{1 + \frac{2\mu}{W_a} s + \frac{s^2}{W_a^2}} \]

with

\[
\begin{align*}
W_a^2 &= \frac{\eta \eta_2}{E_1 E_2} \\
\frac{2\mu}{W_a} &= \frac{\eta_2}{E_1} + \frac{\eta_2}{E_2} + \eta_1 \\
K &= \frac{\eta_2}{E_1}
\end{align*}
\]

For the global model, it is important to distinguish two sub-models \( H(s) \) and \( H'(s) \). The sub-model’s equations are present in the following equivalent form:

- Sub-model 1 - \( H(s) \)

\[ H(s) = \frac{(1 + Bs)A}{1 + Ds + Cs^2} \]  

- Sub-model 2 - \( H'(s) \)

\[ H'(s) = \frac{(1 + B's)A}{1 + D's + C's^2} \]  

By integrating two feedbacks \( FB_1 \) and \( FB_2 \) (J.-Ch. Gille 1971), (R. Laurent. 1985) on the models and by developing preceding equations in order to express the outputs S1 and S2 in function of the inputs E1 and E2, the following system is obtained:

\[
S_1 = \frac{-A(Bs + 1)(Cs^2 + Ds + 1)}{(Cs^2 + Ds + 1)(Cs^2 + D's + 1) - AA FB, FB_1 (Bs + 1)(Bs + 1)} E_1 + AA FB, FB_1 (Bs + 1)(Bs + 1) E_2
\]

\[
S_2 = \frac{-A(Bs + 1)(Cs^2 + Ds + 1)}{(Cs^2 + Ds + 1)(Cs^2 + D's + 1) - AA FB, FB_1 (Bs + 1)(Bs + 1)} E_1 - AA FB, FB_1 (Bs + 1)(Bs + 1) E_2
\]

The model described above contains 10 parameters: A, B, C, D, A, B', C', D', FB1, FB2. The identification of all these parameters requires a use of optimization tools like Genetic Algorithms able to identify parameters avoiding local optima and finding the global one. However, the Bi-model structure implies the adapting of the abovementioned evolutionist algorithm and the use of specific identification strategy described in the next section.

**IDENTIFICATION STRATEGY**

Our approach is defined in two stages. The first consists of the global model identification by genetic algorithms. Second is employed to refine the results starting from a property that the global model may be decomposed into two sub-models.

**“DIRECT” METHOD OF IDENTIFICATION**

The genetic algorithms are tools of optimization being based on techniques derived from the genetics and natural evolution: crossings, changes, selection, etc. The genetic algorithms have already a relatively old history since the first work of John Holland on the adaptive systems goes up at 1962 (John Holland. 1962). The work of David Goldberg (D.E Goldberg 1989) largely contributed to popularize them. The general principle of the operation of a genetic algorithm is given in Figure 3.

![General principle of the genetic algorithms](image-url)

A genetic algorithm looks for the extreme of a function defined on a space of data. The quality of the data coding conditions the success of the genetic algorithms. The binary
coding was very much used in the beginning. Real coding from now on is largely used, in particular in the applicative fields for the optimization of problems to real variables. In our example our fitness function is expressed according to the outputs S1, S2, measured and simulated (Figure 4), i.e.:

\[
f(A, B, C, D, A', B', C', D', FB, FB') = \sum_{i=1}^{n}(y_i - \tilde{y}_i)^2 + \sum_{i=1}^{n}(\tilde{y}_i - \bar{y}_i)^2
\]

In order to validate the identification method compared to our model, many tests were realized starting from a data file simulated with the model described previously. The values of the parameters which we chose relatively near to reality are:

\[
(A = 22.7, B = 0.5, C = 2734, D = 38, A' = 31.1, B' = 0.85, C' = 2260, D' = 43.7, FB_1 = 0.0035, FB_2 = 0.0042)
\]

In Figure 5, the fitness value is presented showing that our criterion of error converges correctly towards 0 for any random initial populations. The initial population is taken with parameters values taken with +/- 5% and +/- 10%.

The identification method has then been tested using simulated curve assimilated to an experimental one (with predefined parameters) and simulated curve obtained from parameters identified by Genetic Algorithms based method. In spite of the fitness function close to zero, the values of parameters were unacceptable. After having analyzed the percentage of errors relative to each parameter, compared to their initial value, we noted that some of them could be compensated by other (P. Bruniaux and al. 1994). For example, the summary table of the whole of the results (Table 1) shows that for a value of our fitness function of 0.08665265974543, certain parameters such as B, B', FB1, FB2 can strongly deviate from their real simulated values. This phenomenon of compensation is marked by the weak difference between the simulated curves that have been assimilated to a real test and the simulated curves obtained from identified parameters obtained using Genetic Algorithms optimization. (Figure 6).

Following these problems, we carried out an analysis of the sensitivity curves for all the parameters separately compared to our error criterion. It showed that certain parameters were not sensitive compared to others justifying the phenomenon of parametric compensation. Also, we chose to increase the richness of our input data entered into the process of identification. For that, the splitting of our model into different sub-models enabled us to integrate complementary tests called uni-axial which are specified in the next paragraph.

**“INDIRECT” METHOD OF IDENTIFICATION**

The splitting of the global model into sub-models can be a very important issue in the identification of a very complex process provided that each sub-model is observable. In our case (textile material), the traction tests can be carried out in the weft and warp directions simultaneously (model of biaxial traction), but also in independent ways (model of uni-axial traction in warp or weft). Compared to our global model, that means observing the outputs S1, S2 according to the inputs E1, E2 for the biaxial case. But in a uni-axial case, we observe only one output according to its respective input although the system is in the feed back configuration. From an identification point of view, it is then possible to identify successively, the warp sub-model, the weft sub-model and
the global model in feedback configuration. These three steps were programmed as follows: the first identifies the parameters A, B, C, D of the sub-model H(s), the second identifies the parameters A’, B’, C’, D’ of the sub-model H'(s) and the last identifies parameters FB1, FB2 by considering the other parameters already known. Figures 7, 9, 11 show that the criterion function evolves correctly to 0 for every identification step. The simulated curves that we assimilated to a real test and the simulated curves obtained from the parameters obtained from the GA identification are quite similar for all identification steps (Figures 8, 10, 12).

Figure 7. Evolution of the fitness function for an exploration field of parameters +/- 10% (warp)

Figure 8. Results of identification of the simulated and initial (warp)

Figure 9. Evolution of the fitness function for an exploration field of parameters +/- 10% (weft)

Figure 10. Results of identification of the simulated and initial (weft)

Figure 11. Evolution of the fitness function for an exploration field of parameters +/- 10% (feedbacks)

Figure 12. Results of identification of the simulated and initial outputs (feedbacks)

The value of the identification results are given in the summary table (1). It is possible to note that each parameter converges towards its optimum (direct method).
Table 1. Comparison of two methods

<table>
<thead>
<tr>
<th></th>
<th>Direct method</th>
<th>Indirect method</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>+/-5%</td>
<td>+/-5%</td>
</tr>
<tr>
<td>A</td>
<td>0.94493</td>
<td>0.370044</td>
</tr>
<tr>
<td>B</td>
<td>0.15</td>
<td>0.251101</td>
</tr>
<tr>
<td>C</td>
<td>0.330834</td>
<td>0.884784</td>
</tr>
<tr>
<td>D</td>
<td>0.085263</td>
<td>1.290474</td>
</tr>
<tr>
<td>A'</td>
<td>0.575563</td>
<td>0.657563</td>
</tr>
<tr>
<td>B'</td>
<td>17.76471</td>
<td>5.294118</td>
</tr>
<tr>
<td>C'</td>
<td>3.410619</td>
<td>1.261947</td>
</tr>
<tr>
<td>D'</td>
<td>0.016018</td>
<td>0.016018</td>
</tr>
<tr>
<td>F1</td>
<td>7.785714</td>
<td>0.028571</td>
</tr>
<tr>
<td>F2</td>
<td>6.761905</td>
<td>0.142857</td>
</tr>
</tbody>
</table>

AGNIESZKA CICHOCKA is preparing her PhD at ENSAIT Roubaix in France and at the University of Lodz in Poland since 2004. She is doing her research in the field of dynamical modelling and simulation of fabrics – Using in design process. She participates to industrial projects with Lectra CAD company specialized in garment industry.

PASCAL BRUNIAUX was born in Denain, France in 1959 and went to the University of Lille, where he studied Automatic control and obtained his PhD in 1988. He worked at HEI engineering institute and in 1990 he came at ENSAIT, the textile engineering institute as Associate Professor. He is teaching in the field of mass customization and textile design process using virtual reality tools. He is doing his research in the field of fabric drape modeling and new approach to garment design using 3D CAD tools.

VLADAN KONCAR was born in Belgrade, Serbia in 1962 France and went to the University of Lille, where he studied Automatic control of complex systems and obtained his PhD in 1991. He worked at the University of Lille as Assistant Professor. In 1991 he came at ENSAIT, the textile engineering institute as Associate Professor. He is teaching in the field of virtual reality, computer networks, communicative and intelligent textile structures and system theory. He is doing his research in the field of intelligent textile sensors and actuators, flexible screens and fabric drape modeling for design purposes.

CONCLUSION
This article presents a new biaxial model of traction in which appear two uni-axial interconnected sub-models. The global model incorporates the feedback loops due to the shearing phenomenon in the fabric.
The economic and scientific interest of this modelling is to be able to jointly identify all the parameters of fabric model starting from only uni-axial test. Also, it was necessary to set up an identification strategy specific to the problem. The identification using genetic algorithm showed a weakness in the precision of the results.
This one could be compensated by setting up an identification strategy based on the decomposition of a model in interconnected sub-models. The environment of measurement strongly contributed to the feasibility of this procedure. In the long run, this type of model will be integrated in a fabric simulator. Unknown parameters of this simulator are related to the traction, bending and shearing of fabric. The interest of our method is that parameters of bending and shearing are obtained from the same test. Previously, two tests were necessary.

RÉFÉRENCES
ADAPTIVE MODEL OF THE HUMAN BODY - METHODOLOGY OF DESIGN OF THE MANNEQUIN MORPHOTYPE

Agnieszka Cichocka
Technical University of Lodz
Faculty of Textile Engineering and Marketing, Department of Clothing Science and Technology
Ul. Żwirki 36, 90-942 Łódź, Poland

Pascal Bruniaux
Laboratoire GEnie et Matériaux TExtiles (GEMTEX), UPR ES EA2161
École Nationale des Arts et Industries Textiles (ENSAIT), 9 rue de l’Ermitage,
BP 30329, 59056 ROUBAIX Cedex 01, France

KEYWORDS
Mannequin morphotype, Adaptive model, Human body, Virtual prototyping, Design model.

ABSTRACT

This article presents a method of creation an adaptive mannequin morphotype in a 3D virtual world. Objective of this work is to be able to control the evolutionary morphology of a mannequin by measurement data that are related to morphotypes of human body obtained from a 3D scanner. The first part of the article aims to position this study in a bigger project namely directly design clothing on a virtual body in three dimensions. It could be also called "Interface Man-Clothing". In the second part, we present the representation model of the human body developed by combining results from various interactive sciences such as anthropometry, biometrics... The third part is devoted to installation of this model in a CAD software adapted to the problem. We describe stage by stage our design methodology. At the end, a presentation of some simulation results is proposed in order to validate the model of mannequin morphotype. The value of the parameters was selected accordingly to the reality of ground.

INTRODUCTION

Nowadays, the design prototyping of clothing is initially executed on a computer in 2D (2 dimensions) then it is transformed to 3D to validate it (Maurbert R.) Outcome of this assembly, must be corrected and adjusted because of the transfer errors and incorrect interpretation between the vision 2D and 3D. This stage of validation is tiresome and expensive for the clothes industry companies. Moreover, it strongly reduces the reactivity of creation. The developments which were carried out in the field of the synthesized image can revolutionize the sector of clothing. The assembly and the visualization of clothing on a virtual mannequin in 3D are in the course of finalization. But this industrial concept will lead to a new model of creation proposed of figure 1. They show various stages to be validated at the time of the executing of this model. (Aussel A., Barjonet C.) (Lourdeaux D., 2001). The first stage consists of the acquisition of a 3D human body model (2) using 3D scanner (1). (Arquès D., Janey N, 1990)(Shao J 1992)(Dillard T. 2004). This transformation will have to follow morphology similar to a morphotype of the French due to measurement of French population (3). The second stage is the introduction of a parameterized morphotype (5) according to the measurement tables (4).

This one will be validated according to the error between the technical pattern data (7) of the underwear sector and the results of the setting flat of these same patronages (6) created.
on this mannequin with equivalent size. The following stage is the direct translation of the interface man-clothing (8). It is based on strategic contours to define the fit of clothing. Then, while taking into account this graphic structure and starting from the clothing style data (9) of clothing to be realized, it is possible to create it in 3D while following an optimized and adaptive methodology of design (10). With new, it will be necessary to validate this process of creation according to the standard error of patronage (12) and these same patronages created on this interfaced mannequin. The setting flat of the pattern (11) contributes to this validation. Lastly, the final stage integrates at the same time the data of the mechanical fabric model (12) and its design (14). It makes it possible to visualize drape of clothing (15) following the assembly of the preceding patronages defined in (10). The introduction of the morphotype (5) presents the object of our study and passes by the definition of an adaptive model of the human body.

ADAPTIVE MODEL OF THE HUMAN BODY

The parameterized mannequin morphotype is based on a definite model of surface representation starting from many curves resulting from the sections of a scan human body. The concept of parameterized model morphotype intervenes in the choice and the position of these curves, their parameter settings, the relations between themselves and the data of the measurement campaign. Amongst other things, this model must join perfectly a methodology design digitized in a CAD to be able to exploit it. The model is managed by various types of parameters in direct relation with the curves controlling the surface world.(Soe at al.,2004) These parameters aim to manage the morphology of the model 3D and its dimensioning. This morphology can vary in function of:

1) morphological parameters managing the length of the members or height of the person,
2) morphological parameters managing volume,
3) parameters of order,
4) parameters of volume distribution between the front and the back of a person.

Morphological parameters managing the length

In order to respect the concept of the beauty (Olivier 1967), (Schreider 1937), (Chiappetta, 2000) the analysis and the introduction of a morphotype requires knowledge of the same time on the respect of the forms and the respect of the proportions of the human body. The study of aesthetics is a science in oneself called anthropometry showing a strong connotation with the concept of the beauty. Pythagore, philosophical austere, recommended a mystical theory of the numbers. He or its pupils discovered a curious relation of harmony between two dimensions: "greatest dimension is with smallest dimension, as the sum from these two reported to largest". This relation is called the Golden Section equals to 1,618. Applied to our model of human body representation, many proportions make it possible to find the Golden Section: the stature to the height of the umbilical point, the stature minus the height of the umbilical point to the width of the shoulders, etc (all = 8/5). The current anthropologists would reverse this ratio and would write 5/8 or 62.5%. Nevertheless, all these proportions were extremely studied and are compared with the stature: the head = 13.5% of the stature, the size postones chest = 30.6%, the lower limb measured until the crotch = 51.3%, the length of the foot = 14.9%, the upper limb = 45.5%, the hand = 10.9%, the index brachial = 74.7%. Several of these proportions are in direct: bond with "the golden section". The modern canon (the canon is a specific limit, a rule) was established scientifically and is based on the anatomy and the average proportions of the human body. "Le canon des Ateliers" is due to Charles Blanc, but other anatomists used it: Quetelet (1871), Topinard (1885), Richer (1890). The important concept which emerges from all this work shows that "the beauty" is very close to this standardized rule, i.e., the average. This concept was checked by the famous Italian biologist Viola, after it worked out and drew the diagram of the body of the average Italian respecting of the quite precise proportions on the whole of these measurements. It noted that the proportions of this morphotype are found in the statues of antiquity. Most perfect is certainly the canon of 7 heads ½. Lysippe, famous for its Apollo de Belvédère, uses like measuring unit the addendum, which one finds 8 times in the measurement of the stature of the body. This old approach is always of topicality if the proportions of a clothing. It makes it possible to identify "strategic curves" which will be extracted from the body scan and which will model the morphology of a human body. By the expression "curves strategic", we think of the limits defined by the standards Europe CEN known under the expression "the primary measurements and secondary measurements". The manufacturers of wood mannequin decided to choose the model of 8 heads because it has the advantage of simplifying the process of creation of clothing (Cooklin1990),(Bily1991).

![Figure 2. Approach the canon of 8 heads related to the patterns definition.](image-url)
translating the concept beauty of the model. These parameters characterize the morphology of each muscle and contribute to the concept of morphotype. They depend on the preceding parameters.

**Morphological parameters managing volume**

These dimensional parameters must be in direct relationship to the data of the measurement campaign and make it possible to control the volume of a human body automatically. Human biometrics (anthropometry) is a part of the biology which analyzes the measurements made on the man starting from statistical laws. By measurements carried out on the man, one will concentrate here to the anthropometrical measures. Two important mathematical laws dominate our problem and are with the source of many practical applications:

1) law of distribution – the values collected in human biometrics have variable frequencies which are distributed according to various laws: laws of Gauss, law of Poisson, etc... En morphology, one meets only the law binominal Gauss.

2) law of correlation - In the human body, all is dependent at least. The form of a part is related often to another. When a dimension increases at the same time as another, it is said that they present a direct correlation (positive); on the contrary if it decreases when the other increases, there is opposite correlation (negative).

Knowing that the variation of the dimensional parameters of a body perhaps represented in the shape "of a normal curve", there exists a linear relationship between them. The observation of a table of measurement confirms this analysis.

This table shows a very significant correlation between the parameter chest girth and the other contours taken in the same direction. The laws of evolution between each parameter represent the anthropometric relations and depend on the type of parameter. Thus, if one reference to the parameter chest girth, a simple relation is written between the other parameters and the chest girth:

\[ y = A_1 \times x + B_1 \]

in which:

- \( y \): value of the other parameters (waist girth, hip girth...),
- \( x \): value of the chest girth,
- \( A_1, B_1 \): constant for a given morphotype.

In the case of a size of break, for certain parameters, the relation is written:

\[ y = A_2 \times x + B_2 + A_3 \times (P_e - x) \]

in which:

- \( P_e \): value of the chest girth for the size of break,
- \( A_2, B_2 \): constant for a given morphotype,
- \( A_3 \): constant controlling the relative variation enters the line of the average slope and the true lines of evolution of the sizes.

**Parameters of control**

The significant number of parameters requires setting up an adaptive procedure to control this model of prediction with a minimum of parameters of entry. To some extent, the morphological parameters reveal two types of variables: internal variables and variables of order. The variables internal are definite and correlated according to the rule of the golden section and the anthropometric relations whereas the variables of order, (Bily 1991), (Cooklin1991) must follow certain rules of evolution and proportion in a 3D space, i.e.;

1) to have the most important dimension in the direction which it controls;
2) to be placed in two perpendicular directions;
3) to be strongly correlated with the parameters which are in the direction of control.

By taking these remarks in consideration, the adaptive human body can be controlled by the parameter stature in the vertical direction and by the parameter contour of chest (relating to the size: 34, 36, 38,...) in the horizontal direction (somebody upright in front of us).

**Parameters of volume distribution between the front and the back of a person**

Studies showed that the mass related to the front of a person same as that is not carried on the back, which brought to define a law of distribution between the front and the back (Seitz at al,2000). A plan of distribution makes it possible to manage this concept. It is defined in order to guarantee that before accounts for 62.5% of the total volume of the body and that the back accounts for 37.5%. This facial plan is positioned perpendicularly with the horizontal section of the chest by respecting these proportions. The intersection of this plan and the symmetry plane of the human body define a homothetic axis. This axis makes it possible to evolve homothetic of various body contours according to the laws of volume distribution managed by the parameters preset in “Morphological parameters managing volume”

**METHODOLOGY OF DESIGN MANNEQUIN MORPHOTYPE**

The tool of digitalization used to extract the relevant data of the human body is the body scanner of company LECTRA. The interest of this system is its direct relation with the data-processing tool Design Concept 3D in which the model of representation of the human body was developed. Then, we pass to the importation of this mannequin morphotype in this software (figure 3a). This mannequin determines the general appearance (morphology) future parametric mannequin,(Wang 2005). This means that it was selected among a whole of human character respecting measurements of the basic size for a given population. It is important to symetrize it upstream this stage in order to eliminate the defects from posture vertical as the figure 3b shows it. This defect can be corrected in the software Design concept 3d. In this position, the reference mark of work must be defined such as plan XY is in front of the user in low position.
The data input which are still useful for us on the mannequin morphotype are the parameters of measurement of each primary education and secondary contour (figure 6). This stage of the process, it is possible for us to create the parametric mannequin morphotype. For that, we extracted by an associative translation various primary education contours and secondaries (figure 7).

Figure 3. Mannequin morphotype

To define the position of primary education contours, it is necessary to set up a database in which we create the control parameters of the mannequin morphotype (figure 4a: foot, knuckle). These parameters represents the position of each section according to stature ha (figure 4b).

Figure 4. Parameters of order & mannequin morphotype with its primary contours

Preceding contours are very useful for the construction of clothing. But, if one wishes to respect the morphology of a human body, we must create other secondary contours. They are oblique in the case of a junction of leg or arm, or horizontal to refine the representation of the muscles (figure 5a). The parameters are also managed per ha (figure 5b: genou1, genou2...).

Figure 5. Secondary contours and their parameters

The following stage makes it possible to symetrize the mannequin. Using properly positioned symmetry plane we create the half contours of each preceding contour (figure 8a). The latter are then symetrized and sew again to define new symmetrical contours (figure 8b). This operation is carried out only for the trunk.

Figure 6. Primary and secondary parameters

Figure 7. Extraction of primary education and secondary contours by translation

Figure 8. Half contours and symetrized contours of the trunk

A delicate stage was to assemble one of the members (the leg) to the trunk. Oblique secondary contours make it possible to take into account these connections of the
kneecap type which one finds in the case of a dynamic mannequin. A contour of the type connection trunk leg is created (figure 9b). For that, a half contour was defined from one of oblique contours and two lines of junction binding oblique contours (figure 9a). This last was then symetrized and we were pressed on these two new elements to create final oblique contour (figure 9c).

Figure 9. Creation of secondary contours

The distribution of the back/front was defined compared to the facial plan (Figure 10a) associated a line of distribution. This line is delimited by the points of end of the half contours of the trunk (Figure 10b). Information resulting from the parameters of measurement of the half contours made it possible to create a point of distribution on this line positioning this plan of distribution.

Figure 10. Distribution back/front

In order to dissociate the mannequin morphotype and the parameterized mannequin, it is necessary to relocate on the plan the whole of useful contours (Figure 11a). This procedure enables us to reposition them according to a new table of parameters of control (figure 11b) managed by the new stature H.

Figure 11. Parameterized dissociation of mannequin morphotype

Figure 12 shows at the same time contours resulting from the mannequin morphotype and new contours of the mannequin parameterized by a stature H different of Ha.

Figure 12. Contours control by H

This stage of the design, we can integrate the anthropometric rules in the form of anthropometric parameters (figure 13: TENC, Mtoc...). Using a central homothetic, whose center is at the medium of the two report/ratio and distribution and symmetry planes, parameter anthropometric/parameter measured (ex: TPOI/poi), we create contours associated with a new size managed by the turn with chest P (figure 14).

Figure 13. Anthropometric rules

Figure 14. Contours associated with a new size
Surfaces of representation of the adaptive mannequin are pressed on the curves of contours of the trunk and the leg separately and use a surface function adapted (gauge) to define the volume of the mannequin. The leg will be duplicated symmetrically (figure 15).

Figure 15. Special surface function (gabarit)

Crotch is designed by creating a surface limited by contours of separate crotch and external contour of these two contours (figure 16)

Figure 16. Creation of between legs contours

**SIMULATION RESULTS**

Figure 17 shows us some results of simulation of this parameterized mannequin morphotype. It is noticed that according to the stature (1657, 1657, 1750, 1800), of the turn of chest (950, 1000, 1000, 1100) or of the combination of these two parameters of entry of the process of creation, the morphology of the mannequin is rather close to sound morphotype and this in spite of the strong imposed variations. The values of the variables are given by two interactive control relations.

Figure 17. Parameterized mannequin morphotype

**CONCLUSIONS**

This article presents a methodology of design of a parameterized mannequin morphotype. The results show that the virtual world is increasingly close to the real world by this principle of creation. Moreover this type of mannequin can be easily imported to the CAD software with VRML files. The anthropometric relations can very easily adapt to the data of the measurement campaign by the means of the constants which they integrate. It still remains to validate this mannequin according to the technical data of patronage of the sector of the linen room and the results of the setting flat of these patronages created on this mannequin. Future work will be to insert complements into this method to take into account deformation of the spinal column involving a back uneven or cambered. The stage describing the interface man-cloth is in the course of creation.

**REFERENCES**

Arquiès D., Janey N., Modélisation 3D de paysages sur micro-ordinateurs, Actes de IASTED, International Symposium on Modelling, Simulation and Optimization, Mai 90, Montréal, Canada, 158-161

Aussel A., Barjonet C., Etude des styles du mobilier, Edition DUNOD

Bily-Czopowa M., Mierowska K. Krój i modelowanie odzieży lekkiiej, Warszawa, 1991


Chiappetta J. La coupe à plat – Le lingerie féminine et le maillot de bain. Les Presses du midi, 1999


Finsterbusch K., Mosinski E. Pohl H. Principes de la construction de vêtements

Geurlain P. Contribution à la métrologie automatique 3D du corps humain.


Maubert R., Anatomie du meuble, Edition VIAL

NF EN 13402-1, Désignations des tailles de vêtements - Partie 1 : termes, définitions et procédés de mesurage du corps, Juin 2001.
AGNIESZKA CICHOCKA is preparing her PhD at ENSAIT Roubaix in France and at the University of Lodz in Poland since 2004. She is doing her research in the field of dynamical modelling and simulation of fabrics – Using in design process. She participates to industrial projects with Lectra CAD company specialized in garment industry.

PASCAL BRUNIAUX was born in Denain, France in 1959 and went to the University of Lille, where he studied Automatic control and obtained his PhD in 1988. He worked at HEI engineering institute and in 1990 he came at ENSAIT, the textile engineering institute as Associate Professor. He is teaching in the field of mass customization and textile design process using virtual reality tools. He is doing his research in the field of fabric drape modeling and new approach to garment design using 3D CAD tools.
SUPPLY CHAIN MULTI-OBJECTIVE SIMULATION OPTIMIZATION

Jeffrey A. Joines
Kristin A Thoney
Department of Textile Engineering
North Carolina State University
Raleigh, NC 27695-8301, U.S.A.

Russell E. King
Michael G. Kay
Department of Industrial Engineering
N.C. Carolina State University
Raleigh, NC 27695-7906, U.S.A.

ABSTRACT

A critical decision companies are faced with on a regular basis is the ordering of products and/or raw materials. Poor decisions can lead to excess inventories that are costly or to insufficient inventory that cannot meet its customer demands. These decisions may be as simple as “How much to order” or “How often to order” to more complex decision forecasting models. This paper addresses optimizing these sourcing decisions within a supply chain to determine robust solutions. Utilizing an existing supply chain simulator, an optimization methodology that employs genetic algorithms is developed to optimize system parameters. The performance measure that is optimized plays a very important role in the quality of the results. The deficiencies in using traditionally used performance measures in optimization are discussed and a new multi-objective GA methodology is developed to overcome these limitations.

1 INTRODUCTION

Competitiveness in today’s marketplace depends heavily on the ability of a company to handle the challenges of reducing lead-times and costs, increasing customer service levels, and improving product quality. Traditionally, sourcing (procurement), production, distribution and marketing have been working independently. Unfortunately, although they seem to be working towards a common goal, these organizational units have different objectives. Marketing wants to have a high customer service level as well as high sales volume, but this conflicts with the objective of production and distribution. Sourcing decisions normally depend solely on minimizing the cost of goods, and production and distribution decisions often consider only maximizing throughput while minimizing production (unit) costs without any consideration for high inventory levels or long lead-times. Supply chain management is the effective coordination and integration of different organizations with different objectives towards a common goal. The great potential for improvement in these objectives through effective supply chain management mechanisms has recently been realized (Karabukal et al. 2000; Lyon et al. 2001).

A supply chain, from an operations perspective, has three components: sourcing or procurement, manufacturing and distribution, and inventory disposal. The focus of this paper is on decision making in the sourcing component. In particular, we develop a simulation-based, genetic procedure for determining optimal setting for controllable inputs. Sourcing decisions have a large impact on manufacturing and distribution and inventory disposal as well. Therefore, sourcing and procurement decisions directly affect the efficiency of the entire supply chain. Because sourcing decisions include the supplier, which is usually a separate company, these decisions are much more rigid than manufacturing, distribution, and inventory disposal decisions. Whereas manufacturing and inventory disposal decisions might be internal to a company most of the time and therefore easier to change or modify, sourcing decisions that include outside companies will be hard to change due to contracts and agreements.

Supply chain problems are often very large and complex owing to the interactions between the entities, the length of the supply chain, the lead times of manufacturing and shipping, the complexities of modeling the individual entities, the stochastic nature of the demands, etc. Because of these complexities, very few analytical models exist except for simplified versions of the problem which often are based on limiting assumptions. Even if the analytical forms do exist, it is very difficult to solve these models using traditional search methods like linear programming, differentiation, or even local gradient-based methods owing to the fact that most of the models are discrete, non-linear and/or multi-modal. Therefore, heuristic or computational methods are required to even determine good solutions.

Computer simulation is a methodology that can be used to directly model the complexities of the entire supply chain without the limiting assumptions. It can be used to describe and analyze the behavior of a supply chain and aid in the design/control of the supply chain through evaluation of “what if” questions (i.e., what if we source from these two suppliers? or what if
we drop ship 20% of the estimated demand from one supplier and quick replenish every four weeks from the other supplier?). However, other practical questions (such as Which combination of suppliers is best? and What is the best sourcing strategy under these conditions?) seek optimum values for the decisions variables of the system for the one or more performance measures. In this case, the simulation model can be thought of as an objective function and/or constraint functions in optimizing these complex stochastic systems.

Using simulation in the optimization process presents several challenges. First, there is no analytical expression of the objective function, which eliminates differentiation, or exact calculation of local gradients. Further, the stochastic nature of the simulation causes problems because given a set of deterministic decision variables, the performance measure is not crisp but rather is described by a probability distribution. Simulation programs are typically computationally more expensive to evaluate than analytical functions. Therefore, the efficiency of the optimization algorithms is more crucial.

### 1.1 Performance Measures

An important issue in supply chain management is how to measure the performance of a supply chain for a given set of decision variables. No matter how appropriate the methodology, if the performance measure is poor, the results could be misleading or false. A key issue is that multiple objective problems are faced frequently in supply chain management (i.e., companies want to optimize more than one criteria that are often conflicting). Typically, the objectives are to maximize revenue with minimal inventory while maximizing customer service. Multi-objective optimization problems are a major challenge in developing solution methodologies. Ideally, one would like to have a single performance measure that addresses all three of these issues. Unfortunately, traditionally used performance measures do not take into account all three. Gross Margin (GM) is a widely-used supply chain performance measure that takes into account the profitability of the company:

\[
GM = \text{Total Revenue (TR)} - \text{Cost} = TR - (# \text{ of units purchased} \times \text{Whole cost/unit}).
\]

GM is a measure of revenue only. It neither takes into account the cost of carrying inventory and nor relates to the chain’s customer service level. In an effort to improve GM, Gross Margin Return on Investment (GMROI) adjusts GM for the average inventory held over the period and is an effective measure that takes into account the money earned and inventory held as calculated in the following formula:

\[
\text{GMROI} = \frac{GM}{\text{Average Inventory Cost}}.
\]

It can be interpreted as the margin earned per dollar invested in inventory. It is one of the measures that is used to evaluate the effectiveness of solution alternatives in the current decision support system built on top of the Sourcing Simulator. However, it does not consider the customer service level and in earlier studies it is shown to be a poor objective function when used directly in an optimization algorithm (Gokce 2002). GMROI is misleading in that it can be maximized by buying just a very few items, in which case GM will be very low but the average inventory will be even lower. Therefore, GMROI is high but the customer service level is very low since only a little of the consumer demand is met while the revenue is also poor. A new measure (GMROISL) was proposed that incorporates customer service by multiplying GMROI by the in-stock percentage (Gokce 2002). However, this presumes a weighting between in-stock percentage and GMROI which is difficult to determine. Also notice that, in measuring customer service, lost sales are more important than the in-stock %. Minimizing the lost sales directly relates to maximizing customer service, while being out of stock for a SKU (Stock Keeping Unit) does not necessarily decrease customer service level in a particular period if there was no demand for that SKU during that period.

### 1.2 Sourcing Simulator

The Sourcing Simulator is a stochastic simulation model for the consumer product retailing process developed by Nuttke et al. (1991) and Hunter et al. (1992). The model allows investigation of the effects of alternative retailing procedures on financial and other performance measures for a retail store. The value of the model lies in the fact that it captures the random nature of consumer behavior at the retail store within a robust framework that allows investigation of buyer strategies. Consumer arrivals at the retail store are modeled as a time-dependent Poisson process. The rate each week is based on a specified season arrival pattern. The model tracks the inventory by SKU. A forecast of consumer demand is expressed in terms of consumer volume, SKU mix and presumed seasonality. The model assumes that this forecast is in error. This error is specified as a volume error, SKU mix error, and actual seasonality.

It models alternative mechanisms for supplying product to a retail store. The model tracks the inventory of a line of product offered in a range of SKUs. The store sets up an initial inventory to start the selling season according to the store buyer’s plan. Customers arrive at the store and attempt to purchase garments. For a particular customer, if the desired SKU is in stock, a sale is recorded and inventory decremented. If the SKU is out of stock, a stockout is recorded. In either case the customer may look for another item with certain probabilities. The store may issue replenishment orders on the vendor. Replenishment may be based upon the
original buyer’s plan or may reflect the use of actual Point-of-Sale (POS) data. In this way the selling season is played out and performance statistics are computed. Within the season, the buyer may employ one of two alternative techniques for re-estimating season’s demand and incorporating the re-estimate in a scheme for issuing reorders to the manufacturer.

The Sourcing Simulator can simulate sourcing scenarios that contain a large number of SKUs. The number of decisions that can be made by the analyst is large and even finding good decisions is very hard in a trial an error approach which is what is currently being done by hundreds of companies that use the tool. Owing to the stochastic nature and non-linearity of the problem as well as the fact that some of the variables are discrete, a solution methodology using GAs will be developed.

2 SIMULATION OPTIMIZATION

Law and McComas (2000) define simulation optimization as the “orchestration of the simulation of a sequence of system configurations (each configuration corresponds to particular settings of the decision variables (factors)) so that a system configuration is eventually obtained that provides an optimal or near optimal solution.” Several excellent survey papers are written on simulation optimization techniques and procedures. Androdottr (1998) and Fu (2001) present reviews of simulation optimization techniques both for continuous and discrete decision variables. A detailed description of available simulation based optimization packages, their vendors, and the heuristic search procedures that they use may be found in Law and Kelton (2000).

2.1 Genetic Algorithm

Since the supply chain optimization problem is very complex, the solution space is not easy to search owing to the landscape and the size of the space (i.e., stochastic simulation). Genetic algorithms (GAs) are a powerful set of stochastic global search techniques that have been shown to produce very good results for a wide class of problems. GAs can find good solutions to linear and nonlinear problems by simultaneously exploring multiple regions of the solution space and exponentially exploiting promising areas through mutation, crossover and selection operations (Michalewicz 1996). In general, the fittest individuals of any population are more likely to reproduce and survive to the next generation, therefore improving successive generations. However, some of the inferior individuals can, by chance, survive and also reproduce. Unlike many other optimization techniques, GAs do not make strong assumptions about the form of the objective function (Michalewicz 1996). The general GA is summarized in Figure 1.

1. Set generation counter $i \leftarrow 0$.
2. Create the initial population, $P_0$, by randomly generating $N$ individuals.
3. Determine the fitness of each individual in the population by applying the objective function to the individual and recording the value found.
4. Increment to the next generation, $i \leftarrow i + 1$.
5. Create the new population, $P_i$, by selecting $N$ individuals stochastically based on the fitness from the previous population, $P_{i-1}$.
   
   (a) Randomly select $R$ parents from the new population to form the new children by application of the genetic operators.
   
   (b) Evaluate the fitness of the newly formed children by applying the objective function.
6. If $i < \text{the maximum number of generations to be considered}$, go to Step 4.
7. Output the best solution found.

Figure 1: A Simple Genetic Algorithm

2.2 Multi-objective GA

Often optimization problems often have multiple objectives. Most of the time these objectives are conflicting (i.e., optimizing one objective causes the other objectives to be poor). For example, consider a grocery store simulation where one is trying to determine the optimal number of baggers and check out clerks needed during each time period the grocery store is open. Minimizing the overall cost will ultimately lead to only one bagger and clerk for each period while optimizing for the minimum customer wait time would lead to $n$ baggers and clerks where $n$ is the number of checkout lines. When only looking at one objective, the other objective suffers. However, in this case both objectives are quite important and they need to be optimized together.

In this paper, the objectives ($f_1, f_2, \ldots, f_k$) are output performances generated from a stochastic simulation, making the search even more difficult. It is very rare that a single optimal point will optimize all of the objectives at once. Therefore the notion of optimal is different than when considering a single objective, the term Pareto optimal will be used. A solution $i(X_i)$ is Pareto optimal if there does not exist another point $X_2 \in F$ such that $f_i(x_1) \leq f_i(x_2)$ for all $i = 1, \ldots, k$ except $i = j$ and $f_j(x_1) < f_j(x_2)$ for at least one $j$. A set of non-dominated solutions is now obtained to generate a Pareto optimal frontier.

2.3 Multi-objective Optimization

Over the past 20 years, many traditional methods have been developed for this problem. Most of these methods are quite limiting in their abilities to solve only linear objectives and constraints. Since the objectives are being generated from a stochastic simulation, the methods appropriate for simulation optimization will have to be
used. See Coello Collelo (2002) for a critical review of all evolutionary multiobjective techniques. There are three approaches that one could use to solve the objective value.

2.3.1 Decision before Search

Probably the simplest and easiest method is to aggregate the objectives into a singular scalar function

$$g(x) = \sum_{i=1}^{k} w_i f_i(x)$$

where $w_i$ are the weighting functions which can determine the relative importance of each of the objectives and $\sum_{i=1}^{k} w_i = 1$. The relative importance is determined by the utility of the decision maker. Now any of the standard scalar optimization techniques can be employed to solve it (e.g., Stochastic Approximation, OptQuest, etc.). The solution determined represents only one point on the Pareto frontier. Multiple set of weight values can be tried but there is no guarantee the frontier can be generated. Also, determining the relative weights can be quite difficult since the scale of the individual objectives plays a dramatic role. For example, the case of trying to maximize GMROI (a cost metric) and service level percentage, GMROI will dominate the objective to effectively only optimizing GMROI.

Other approaches include goal programming where the decision maker provides targets or goals that they wish to achieve for each objective. The purpose is to try to minimize the absolute deviations among the objective and the targets ($\sum_{i=1}^{k} |f_i(x) - T_i|$). Again, single objective techniques can be used. However, the decision maker has to know about the individual objective ranges and the solution generated will lie in the dominated region if the targets do.

Another more effective method of using thresholds, is to optimize over one objective ($f_1$) and constrain the other objectives to be within some threshold. For example, one could maximize GMROI while constraining the service level % to be at least 95%. There are two criticisms of this approach for decision making. One, under certain conditions (variable ranges, data) 95% may not be obtainable. Second, this will be a binding constraint and will produce only one point on the frontier.

2.3.2 Search before Decision

When producing only one point, the decision portion has been done before the search process has been performed. In the previous example, the possibility that GMROI could be doubled if the company were willing to accept a 92% service level would be lost in the analysis. Having the frontier allows for a more robust decision to be made since the frontier gives the analyst the ability to see all possibilities. Pareto based GA methods try to generate the entire Pareto frontier which will allow the decision maker to make a decision based on the frontier. Most of the GA Pareto methods modify the selection process in determining which points survive as well as undergo reproduction to generate new points (Coello Coello, 2002). The modifications take into account that there are more than one objective.

Vector Evaluated GA (VEGA) by Schaffer was the first multi-objective GA method. It modifies the selection operator by performing proportional (roulette) wheel selection using each objective to select a number of sub-populations. For example, if there are two objectives, half the population will be selected using $f_1$ and the other half using $f_2$. Then the sub-populations are shuffled together to form a new population. This is very simple and efficient, but solutions generated are what are called locally nondominated but not necessarily globally nondominated. Individuals excel only along one objective. The frontier created will be mainly clustered near the ends of the frontier (as if we only optimized one objective). More efficient methods like the Niched Pareto GA (Horn et al., 1994) more effectively generate a full Pareto frontier.

Nondominated Sorting GA (NSGA-II) by Deb and Goel (2002) is one of the best methods for generating the Pareto frontier and is used in this study. The NSGA-II algorithm ranks the individuals based on dominance. The fast nondominated sorting procedure (Figure 2) allows us to find the nondomination frontiers in Step 4 of the main algorithm where individuals in the frontier set $F_1$ are not dominated by any solution and those in $F_2$ are dominated by only those individuals in $F_1$. Also, it determines a new fitness value ($v_{rank}$) that represents the frontier number (i.e., all individuals in $F_1$ are given a fitness of one, $F_2$ two, etc.).

In Step 6a of the main algorithm (Figure 5), the crowding distance is calculated (seen in Figure 3) for each individual of the new population. Crowding factor gives the GA the ability to distinguish individuals that have the same rank (i.e., those that reside in the same frontier set $F_i$). This forces the GA to uniformly cover the frontier rather than bunching up at several good points by trying to keep population diversity.

The comparison operator ($\geq_n$) given in Figure 4 is used by the GA to sort the population for selection purposes. Individuals that are in a lower domination frontier set are considered better than those in higher sets (i.e., $F_1$ individuals are relatively better than individuals in $F_2$). If they are in the same frontier, then the individuals which is the farthest from other individuals is considered better (i.e., this individual fills a gap on this frontier set).

The NSGA-II main loop can be seen in Figure 5. The first step is the same as Steps 2 and 3 of the general GA in Figure 1. In Step 3, we add the newly generated children ($Q_t$) to the parent population. Next, the domination frontiers are determined as well as the new fitness
1. For each $p \in P$
   (a) For each $q \in P$
      - If $p$ dominates $q$ then $S_p = S_q \cup \{q\}$ which includes $q$ into set of solutions domin. by $p$
      Else if $q$ dominates $p$ then
        $n_p = n_p + 1$
   - If $n_p = 0$ then $F_i = F_i \cup \{p\}$
2. $i = 1$
3. While $F_i \neq \{\}$
   - $H = \{\}$
   - For each $p \in F_i$
     (a) For each $q \in S_p$
       - if $n_q = n_q - 1$
         - $H = H \cup \{q\}$
     - $i = i + 1$
     - $F_i = H$ next frontier has now been formed
4. Return the set of frontiers $F_i$

Figure 2: Fast Nondominated Sorting

1. $i_{distance}(j) = 0 \forall j = 1, \ldots, p$
2. For each objective $k$,
   - Sort $P$ based on $k$
   - $i_{distance}(1) = i_{distance}(p) = \infty$
   - Boundaries are always selected
   - For $j = 2$ to $p - 1$
     $i_{distance}(j) = i_{distance}(j) + (f_{m}(i + 1) - f_{m}(i - 1))$

Figure 3: Crowding Distance Function

- Given the non-domination rank ($i_{rank}$) and local crowding distance ($i_{distance}$)
- $i \geq n$ if
  $(\text{rank} < j_{\text{rank}} \text{ or } ((\text{rank} = j_{\text{rank}} \text{ and } (i_{\text{distance}} > j_{\text{distance}})))$

Figure 4: Comparison Operator ($\geq_n$)

1. Randomly initialize $P_0$ and set $Q_0 \leftarrow \{\}$
2. $t \leftarrow t + 1$
3. $R_t \leftarrow P_t \cup Q_t$ combine parent and children population
4. $[F_t, i_{\text{rank}}] \leftarrow \text{fast_nondominated_sort}(R_t)$ where $F$ equals all non-dominated fronts of $R_t$
5. $P_{t+1} \leftarrow \{\}$
6. While $|P_{t+1}| < N$
   (a) $i_{\text{distance}} \leftarrow \text{crowding_distance_assignment}(F_t)$
   (b) $P_{t+1} = P_{t+1} \cup F_t$ include the $i$ dominated set in next population
7. Sort($P_{t+1}, \geq_n, i_{\text{rank}}, i_{\text{distance}}$)
8. Select the top $N$, $P_{t+1} = P_{t+1}[1 : N]$
9. $Q_{t+1} \leftarrow \text{make_new.pop}(P_{t+1})$ via selection, crossover, and mutation
10. If stopping criteria met then stop otherwise go to 3

Figure 5: NSGA-II Main Algorithm

to modify NSGA-II algorithm to work in this environment to allow them to see the frontier so they could make their own decisions. Table 1 represents the fixed scenario specifics as defined by the company.

Table 1: Scenario Specifics

| Planned number of units to sell               | 4800  |
| Season Length                                | 28 weeks |
| Sensibility                                  | Mill peak |
| No. of Styles/Colors/Bases                   | 2/4/0 (48 SKUs) |
| Planned Forecast                             | 30/30/40 % |
| Style Color                                  | 15/15/15/15/15/15 % |
| SKU Mix Error                                | 0.0 % |
| Style Color                                  | 40 % |
| Color                                       | 40 % |
| Style                                       | 20 % |

<table>
<thead>
<tr>
<th>Cost Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>initial whole sale cost</td>
</tr>
<tr>
<td>replenishment whole sale cost</td>
</tr>
<tr>
<td>retail price</td>
</tr>
<tr>
<td>liquidation price</td>
</tr>
<tr>
<td>ordering Cost</td>
</tr>
<tr>
<td>initial fixed shipping cost</td>
</tr>
<tr>
<td>replenishment fixed shipping cost</td>
</tr>
<tr>
<td>% who choose alternative</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Markdowns</th>
</tr>
</thead>
<tbody>
<tr>
<td>No. of markdowns</td>
</tr>
<tr>
<td>Markdown Start Week</td>
</tr>
<tr>
<td>Markdown Duration</td>
</tr>
<tr>
<td>Markdown %</td>
</tr>
<tr>
<td>% who choose alternatives</td>
</tr>
</tbody>
</table>

3.1 Modified GA

In building a GA methodology to solve the supply chain sourcing problem, six fundamental issues that affect the performance of the GA must be addressed: chromosome representation, initialization of the population, selection strategy, genetic operators, termination criteria,
and evaluation measures. In the following subsections, these issues are introduced and described specifically for the proposed multi-objective GA.

**Chromosome Representation** For any GA, a chromosome representation is needed to describe each individual in the population. Chromosome representation determines how the problem is structured in the GA, as well as the genetic operators that can be used. For the sourcing decision, the chromosome representation in this case is fairly straightforward. An individual is kept as a vector of size 51, where each cell corresponds to a decision variable (as seen in Table 2).

<table>
<thead>
<tr>
<th>Decision Variable</th>
<th># of Variables</th>
<th>Bounds</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Initial Drop %</td>
<td>48</td>
<td>[0 100]</td>
<td>Real</td>
</tr>
<tr>
<td>Number of Reorders</td>
<td>1</td>
<td>[0 18]</td>
<td>Discrete</td>
</tr>
<tr>
<td>Reorder start week</td>
<td>1</td>
<td>[1 18]</td>
<td>Discrete</td>
</tr>
<tr>
<td>Min. Order Quantity</td>
<td>1</td>
<td>[1 60]</td>
<td>Discrete</td>
</tr>
</tbody>
</table>

Notice, that not all combinations of the decision variables constitute a feasible solution. For this reason, infeasible solutions are repaired using a repair function before they are evaluated by the Sourcing Simulator.

**Initialization of the Population** The initial population is formed randomly based on the upper and lower bound for each of the decision variables in a chromosome using a uniform distribution.

**Selection Strategy** Selection of parents to produce successive generations is very important in driving the search. The goal is to give more chance to the “fittest” individuals to be selected. For each selection scheme, probabilities are assigned to the individuals. The better individuals have higher probabilities. A normalized geometric ranking scheme is used for the proposed genetic algorithm in this paper. Individuals are first ranked from best to worst according to their fitness values. Then each individual is assigned a probability based on the rank from a truncated geometric distribution (Joines et al. 1996). In the original NSGA-II method, Deb et al. use a tournament selection where the tournament is based on the $\geq_n$ operator. Since the population is sorted from best to worst in Step 7, the normalized geometric ranking scheme does not require any more sampling or sorting as does the tournament selection.

**Genetic Operators** Reproduction is carried out by application of genetic operators on selected parents. Four mutation (Boundary, Uniform, Nonuniform, and Multi-NonUniform) and three crossover operators (Simple, Arithmetic, and Heuristic) are used based on the representation. Continuous variables use the version by Michalewicz (1996) while the discrete variables use the modifications by Joines et al. (1996).

**Termination Criteria** The GA is terminated after a specified number of generations.

**Evaluation Measure** Genetic algorithms rely on the simple premise of using natural selection as a means of solution elimination. The objective function is the driving the force of the GA search. In this research, instead of performing an analytical function evaluation, each solution is simulated to determine its performance. Because the simulation is based on a particular forecasted demand level and seasonality (i.e., nonhomogeneous process of customers arriving to the store), the answers generated need to be as robust as possible. Therefore, we performed 25 replications at each of the following volume errors (i.e., $-20, -15, -10, -5, 0, 10, 15$, and $20\%$) where $-20\%$ error means that the true demand will be $20\%$ under what was planned. An overall average and standard deviation across all error levels and replications for GMROI and service level % is used as the objective value for those input parameters. Each of the replications uses common random numbers.

### 3.2 Implementation and Results

The Sourcing Simulator code was linked with the GA Optimization Toolbox (GAOT) in MATLAB developed by Joines et al. (1995). This allows the methodology to be tested before embedding it into the decision support system and Fortran simulation code. The algorithm was run for 300, 500, 700 generations and the Pareto Frontiers generated can be seen in Figures 6, 7, and 8. As can be seen the GA does a nice job of generating the frontiers. The longer the GA is run, the more evenly the frontier is covered. Using a 700 MbZ laptop, 300 generations took approximately 20 minutes while 700 generations took 48 minutes.

![Figure 6: Pareto Frontier for 300 Generations](image.png)

4 CONCLUSIONS

For this data, the service level of 95% produced a GMROI of 1.5 while a 92% service level produced a GMROI of 2.75. It can be seen that the GMROI can be doubled
with a reduction to a 90% service requirement. The frontier allows the analyst to make the best decision. In some cases, the GMROI has not changed much based on decrease in service level. The scenario specific results constrained at 95% service level might not be the best option for the retailer. For a fast moving garment one might wish to maintain a 99% service level. For slower moving items, one might be willing to run out of stock, lowering the service level and spending less on inventory. Because the optimization routine works well, the simulation code is being expanded to allow for different service levels based on volume demand. In the current scenario, the number of weeks of supply stored in inventory was constant across SKUs. Currently, investigating optimizing what the weeks of supply should be for each SKU. Also, the robustness of the algorithm needs to be verified (i.e., the number of replications).

REFERENCES


DECISION SUPPORT SYSTEMS
AN OBJECT ORIENTED TIME MODEL OF A DECISION SUPPORT SYSTEM FOR INTOXICATION DIAGNOSES

Joël COLLOC
University of Le Havre, 25, rue Philippe Lebon
BP 540 F-76058, Le Havre, France
LEACM, EA654, ISH, University of Lyon 2
16 , Avenue Berthelot 69363 Lyon Cedex 07 France
Joel.colloc@univ-lehavre.fr

Peter SUMMONS
The University of Newcastle
University Drive, Callaghan, NSW 2308, Australia
Peter.Summons@newcastle.edu.au

ABSTRACT:
This paper presents a method exploiting Object Oriented (O-O) model concepts to build a Knowledge Base System (KBS) in empirical domains such as the Human Sciences. The method relies on a particular model whose specific features are: a twin conceptual level (internal and external level); a multiple ascending inner inheritance based on object composition; evaluation and dynamical functions; and a zoom operator providing an adapted perception of the expert's domain. Model, method, and automated tools yield the design of a KBS for the diagnostic aid of intoxications with antidepressive drugs. It relies on fuzzy functions and time modelling, and takes a case based reasoning approach.

KEYWORDS: O-O model, method, fuzzy function, diagnostic aid, case based reasoning

INTRODUCTION
We propose an O-O model and a method to design KBS in Medicine. Our method is an aid to Decision Support System (DSS) and Knowledge Base System (KBS) development. The method relies on a life cycle and guides the actors from specifications to KBS validation. We illustrate the use of both model and method to elaborate a DSS for diagnostic help of intoxications with antidepressive drugs. The model allows us to build a framework to express the disease concept. The paper concludes with an outline of a KBS design implementation.

1. THE CONTEXT OF MEDICAL APPLICATIONS

1.1. Medical complexity
Medicine is a science based on observation of the human organism. The complexity of human beings forbids an exhaustive representation of organism mechanisms, and moreover of disease genesis. In spite of the progress of science in simulation and modelling, physicians still often have to reason within an uncertain universe.

1.2. The heterogeneous and the experimental nature of medical knowledge.
Medical theories, based on the studies of clinical signs and on experiments, can vary with medical schools. Variations can also arise due to the:
- individual patient - the same disease can express itself in different manners for each patient.
- topography - some diseases are multifocal and show various clinical pictures. For example, bone tuberculosis and pulmonary phthisis are two forms of the same disease.
- periodicity - diseases are dynamic processes, while clinical syndromes are only pathology snapshots corresponding to specific evolution steps of the dynamic process. Some diseases progresses linearly (step by step), for example syphilis displays three successive stages, whilst others, such as a duodenal ulcer or a herpes infection, progress in a cyclic manner. Therefore, time is a major factor when describing pathologies (Colloc and Summons, 2003).
- chosen treatment - an improper treatment or the presence of another underlying disease can interfere and give unusual clinical aspects, as well as possibly producing iatrogenic diseases.
- scientific progress - medical knowledge is continually changing and adapting to new discoveries of researchers.

2. AN OBJECT ORIENTED (O-O) MODEL AND METHOD APPLIED TO DISEASES REPRESENTATION
A KBS needs three components: domain experts, knowledge for reasoning, and a representation model well suited to the concerned domain, and expert reasoning. In this paper, we will focus on the last point.

2.1. Defining a representation for the disease concept
A disease is a concept built according to the experience of many physicians and with the help of medical semiology precepts. Physicians are used to describing a disease by grouping its manifestations into clinical pictures. A principal clinical picture is initially described, forming the skeleton of the disease concept. Then several variations are derived corresponding to frequent occurrences or evolutionary forms.
Each clinical picture is then expressed in terms of syndromes and signs. The medical semiology establishes a classification of the clinical signs observed on the patient. A sign is called pathognomonic whenever its existence must be stated to think about a given syndrome. It is only evocative if its presence strongly suggests the syndrome. Finally, a sign is considered accessory when it is observed to occur in an inconstant manner, or if it is not specific. Therefore, a disease can easily be represented as the root of an aggregation tree whose nodes are clinical pictures at the first level, syndromes at the second level, and signs at the bottom or root level (figure 1).

To obtain a complete model, physicians need means to represent time and to express that the disease displays a different clinical picture for each step of its evolution. So, some nodes of the disease tree can be apparent at one step and disappear at the next evolutionary step. This feature is especially important for intoxications because the evolution is very quick and the treatment must be established as soon as possible to bring the patient through the episode. This feature can be described with interval function operators that represent the creation and continued existence of objects arguments. This work could be based on Allen’s interval operators (Allen, 1984) or on derivations of these operators such as those proposed by Summons (Summons, 2005).

Although rule-based models have been widely used to build many diagnosis KBS in the medical domain, they have criticisms regarding their ability to model deep knowledge as defined by Turban (2005, p581) and their representation of causality (Zahedi, 1993), so we believe that they are not well suited to disease representation. They have difficulties in dealing with medical reasoning areas such as time representation, complex structure evolution, threshold incertitude in evaluating attributes and the triggers of some contradictory rules. Also, interesting medicine applications often need complex knowledge bases and consistency problems between rules will often increase with the size of the knowledge base.

Several works propose a hybrid approach mixing rules and objects but they do not provide any solution for the problems described previously. We chose to use an object-oriented model because it provides disease representations very close to the physician’s medical reasoning process and therefore should facilitate knowledge acquisition in the medical domain.

2.2. The object-oriented model features.

In our model, objects are nested within each other using the composition relationship. According to the application, a unit corresponding to a reference object type (or level 0 object) has to be defined. Each instance of this type is called a unique object. This notion defines a boundary between the internal level and the external level of the reference object. All objects with a bigger, or an equal, size belong to the external environment while the smaller objects are contained in the internal environment of the "unique object". These latter objects are termed sub-objects of the unique object as they contribute to its structure.

The internal structure of the unique object represents the content of the object and its composing sub-objects which determine the partition boundary.

The external environment expresses the relationships of a unique object object with its sub-objects. At any moment, we can zoom in on the sub-objects at this level. This defines the "zoom operator" which allows to fine-tune the granularity of the knowledge objects under consideration. The model provides a twin conceptual level (figure 1).

2.2.1. The internal level or "inner object"

The internal level of a unique object includes: composition of sub-objects (IS-PART-OF) and the relationships grouping all the attributes of an object (HAS-A). Attributes are simple variables that can be wrapped as in Java.

Aggregation relationships establish a multiple upward inheritance which passes sub-objects properties on to form a "compound object". The internal level encapsulates attributes and functions. These represent the static characteristics and behaviour of its component objects. There are constraints on the objects that are either static (constraints on the structures, attributes, and cardinalities), or dynamic (in this case they are functions of time).

2.2.2. The external level or outer-object

The external level of each unique object includes the generalization and/or specialization relationships (IS-A) between object types and object sub-types. These relationships establish a type hierarchy and a simple descending inheritance mechanism which passes superior level object’s properties on to its sub-types (as in the class inheritance hierarchy in Java and C++). The external level contains all the other objects which are not sub-objects of the unique object.

Global evaluation functions belonging to instances of the application object type assess all the object type instances in the knowledge base at a specific reference level. The model distinguishes between type concept (object structure and properties) and class concept (a set of instances).

During instantiation, objects are created according to their defining class constraints (object structure, attribute values and behaviour). In such knowledge objects we define the concept of qualifier attributes as particular attributes (listed as characteristics of the object type) whose instantiation is mandatory at the internal level. Where attributes of an object are not "qualifier attributes" value attribute is optional. The attribute value constraints are verified by input functions. Output functions provide the state of each knowledge object.

The spontaneous evolution of the object’s state is translated by dynamical functions (Fd1 to Fd3) of the inner object environment (figure 2). These functions automatically modify the composition links and the reference values of the attributes, at any time. The unit is the hour. The reference period is 6h.

For example, evaluation functions of the "inner object" compare the structure of an object (based on its composing sub-objects) and the value of particular attributes to those of other unique objects objects. The signs are not represented on figure 2 but they are with the clinical picture objects NS, AS and CS described in section 3. To preserve clarity,
Hematology syndrome and Hepatology syndrome are not represented in Figure 2.

![Diagram](image)

Figure 2: Composition evolution of a disease object

2.2.3 Application object type

The *application object* type is a predefined system type, whose aim is to describe applications which suit with several user needs. Objects of application type encapsulate:
- data metaattributes used to represent intermediate states and results of a treatment or a reasoning step.
- metafunctions that are designed to define existing relationships, other than composition, which exist between knowledge domain objects of some type that are involved in a reasoning step.

In KBS applications, using the zoom effect (figure 1), application objects select the knowledge object types to adapt the appropriate level of reasoning and to trigger evaluation functions. They collect the results of the appropriate comparisons performed between objects of the knowledge base and those input by the user.

2.4. An O-O method to build Knowledge Base Systems

2.4.1. Preliminary steps

An opportuneness study and a feasibility study determine successively the project interest, and the human and material resources needed and available for successful achievement of the project.

2.4.2. The conceptual step

First, the experts, with the help of a knowledge engineer, have to define the knowledge domains concerned with the KBS project, and their relationships. The zoom capabilities of the OO model are exploited to focus the expertise level according to the accuracy required by each knowledge domain. Then, during a domain description stage, the experts must define the characteristics, the structure, and the behavior of instances belonging to the particular object types involved in each knowledge domain. Two kinds of knowledge object types have to be distinguished:

- *concrete object types* that define real world objects (structure and behaviour) according to the perceptions provided by at least one of the five human sensory organs. (For example: a car).
- *abstract object types* that are concepts generated by human cognitive activity. (For example: a disease)

In a KBS, it is not recommended to establish composition links between concrete objects and abstract objects, involving them in the same composition hierarchy. When this rule is respected, the description of object structures may take place in an ascending (top down) or a descending (bottom up) manner.

**Top down design**

For each complex object type of the domain, the experts must determine the number and the type of sub-objects needed to build the inner composition level of their instances. This choice allows us to accommodate the expertise level. To realize the task, it is helpful to select significant knowledge instances. For each knowledge object, experts have to list its sub-objects (and their respective types) and describe each of them in turn, continuing downwards towards the root level of the composition hierarchy, where only simple sub-objects (possessing only their own attributes) are found.

At this stage, the type of a composite knowledge object is expressed by listing the types of its sub-objects involved at each level of the composition hierarchy of its knowledge instances. It is also helpful to create several versions of object types, coexisting without conflict, to represent multiple expert advice when more than one domain expert is involved.

Attribute constraints of each object type are also defined at this step. Finally, the whole object type of each knowledge domain is represented by a diagram and submitted for the domain expert's approval and correction.

**Bottom up design**

Experts build a library of simple object types including their qualifier attributes and respective constraints. Then, the experts use these pre-defined simple object types, whose instances are selected as sub-objects, to build the composition hierarchy of more complex objects. This process continues in an incremental fashion. The resulting composite object type is expressed by listing all types of the sub-objects involved in its composition hierarchy.

The advantages of this approach are the ability to add new sub-object types at any time of the design and the possibility to use pre-existing object types whose quality can be tested before their implication in more complex object types.

Generally however, ascendant and descendant method are combined and a reference dictionary gathers object-type identifiers.
The next stage is concerned with the definition of interface functions, evaluation functions, and dynamic functions, which can express object states and attribute values at any time of the object existence. All of these features are included in their respective object type headers. The resulting domain objects and types constitute the domain knowledge base.

2.4.3. The application description step
During this step, the experts express the information needs for the different users by defining abstract application object types. Application objects interact with objects belonging to one or several knowledge domain of the knowledge base built at the previous step. Application objects express the different reasoning steps (similitude, deduction, etc...) performed on knowledge objects.

2.4.4. The implementation step
The implementation step is one of the classical steps comprising the information system life cycle: It consists of prototyping; control and validation of the knowledge base; actual situation evaluation; and finally maintenance (Harmon, 1990).

2.5. A general framework to build disease objects.
Our approach is to build a composition tree, the root of which (level 0) stands for a considered disease. (figure 1) Composition arcs bind the root "disease object" (level 0) with "clinical picture sub-objects" (level 1). Then, other arcs connect clinical pictures sub-objects to "syndrome sub-objects" (level 2). Finally, syndrome sub-objects are linked with sign sub-objects which belong to one of three sign sub-types: pathognomonic, evocative; or accessory signs. A disease object is defined at each time by:
- composition links between clinical pictures, syndromes, and signs.
- dynamic functions which establish the state and structure of the disease object at any stage of its pathologic evolution.
- Its sub-object properties, represented by attribute values. A given sign type can instantiate sign objects involved in many different syndromes, or clinical pictures involved in various nosologic schemas. This approach avoids reasoning loops and provides inheritance facilities.

3. DIAGNOSES HELP OF ACUTE INTOXICATIONS WITH ANTIDEPRESSIVE DRUGS.
Table 1 briefly provides a representation of the general clinical picture of acute intoxication with antidepressive drugs in an emergency service. It depicts the clinical pictures, syndromes and signs and their evolution during 3 days. Table 2 illustrates the sub-type specialisation of clinical pictures according to some specific drugs effects. In deriving sub-types to express clinical forms produced by some specific molecules, we only specify the differences existing between the general picture and the specific drug picture.

4. THE STRUCTURE OF AN OO KBS
This section briefly describes the nature of the principal modules of a DSS based on an OO representation.

4.1 Similarity functions relying on object composition
We propose a set of composition operators for knowledge objects (disease objects) relying on fuzzy logic (Zadeh, 1965). Let E be the universe representing the domain E of disease objects. In fuzzy logic the degree of membership of an object x in the set A, A C E is described by a membership function f_A(x) defined on the [0,1] interval. This function is based on the combination of a set of inclusion criteria of the object x for it to be a member of the set A. Similarly, we can define f_B(x) concerning the set B C E. min is the minimum operator and max is the maximum operator. Intersection A ∩ B is defined by the function f_{A∩B}(x) = min(f_A(x), f_B(x)), the union A ∪ B by f_{A∪B}(x) = max(f_A(x), f_B(x)) and the complement of A in E by f_A(x) = 1 - f_A(x).

### Table 1 - ACUTE INTOXICATION OBJECT TYPE

<table>
<thead>
<tr>
<th>Initial State : SYNDROMES &amp; SIGNS</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>CIRCUMSTANCES OF DISCOVERY</strong></td>
</tr>
<tr>
<td>brought by emergency services, bottle of pills found</td>
</tr>
<tr>
<td>if adult the toxic dose &gt; 10 to 20 mg/kg</td>
</tr>
<tr>
<td>if children the toxic dose &gt; 5 to 10 mg/kg</td>
</tr>
<tr>
<td><strong>ANTICHOLINERGIC SYNDROME (AS)</strong></td>
</tr>
<tr>
<td>Mydriasis, mouth dryness, urinary retention, tachycardia</td>
</tr>
<tr>
<td><strong>NEUROLOGICAL SYNDROME (NS)</strong></td>
</tr>
<tr>
<td>Agitation, drowsiness, coma, hallucinations [4-48h]</td>
</tr>
<tr>
<td>pyramidal irritation, hypertreflexia, myoclonia, convulsions, respiratory depression</td>
</tr>
<tr>
<td><strong>CARDIOVASCULAR SYNDROME (CS)</strong></td>
</tr>
<tr>
<td>if [1g, 0.5g] : sinus tachycardia</td>
</tr>
<tr>
<td>- repoarisation disturbances</td>
</tr>
<tr>
<td>if &gt; 1.5g conduction disturbances</td>
</tr>
<tr>
<td>- slowing of cardiac rythm, arrhythmia</td>
</tr>
<tr>
<td>if &gt;&gt;=1g sudden death by cardiac arrest</td>
</tr>
<tr>
<td><strong>BIOLOGY PICTURE (BP)</strong></td>
</tr>
<tr>
<td>severe intoxication when blood rate &gt; 1000 ng/l= 1μg/l</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>CURE CLINICAL PICTURE - TIME = [+- 24 h] SYNDROMES &amp; SIGNS</th>
</tr>
</thead>
<tbody>
<tr>
<td>NS : agitated awakening, hallucinations</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>CLINICAL PICTURE - SIDE-EFFECTS - AFTER-EFFECTS - TIME &gt;[48 h] SYNDROMES &amp; SIGNS</th>
</tr>
</thead>
<tbody>
<tr>
<td>AS : visual focusing deficiency, acute glaucoma, mouth dryness, urinary retention</td>
</tr>
<tr>
<td>NS : tremor, dysarthria, seizure</td>
</tr>
<tr>
<td><strong>HEPATIC SYNDROME (HPS)</strong> cholestatic jaundice</td>
</tr>
<tr>
<td><strong>HEMATOLOGY SYNDROME (HMS)</strong> agranulocytosis</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Table 2: CLINICAL PICTURE SUB-TYPES</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>MAPROTELLINE NS</strong> : increase of seizure, CS : ↓ cardiac disturbances AS : ↓ anticholinergic effects</td>
</tr>
<tr>
<td><strong>AMINEPTINE AS</strong> : none, HS : increase [30 + 48 h]</td>
</tr>
<tr>
<td><strong>MIAMSERINE CS</strong> : none</td>
</tr>
<tr>
<td><strong>NAMIFENSINE : NS</strong> : psychostimulant effect</td>
</tr>
<tr>
<td><strong>CS</strong> : ↓ cardiac disturbances</td>
</tr>
<tr>
<td><strong>VILOXAZINE : NS</strong> : inhibition, AS : none</td>
</tr>
</tbody>
</table>
4.2. Similarity functions based on the fuzzy Logic
Let $S$ be a set of simple objects signs in our example.

$$\text{Sim} : S \times S \rightarrow [0,1] \quad \text{Sim}_j(O_x, O_y) = \frac{\sum_{i=1}^{n} w_i \cdot \text{sim}(a_i, b_i)}{\sum_{i=1}^{n} w_i} \quad (1)$$

When $O_x$ and $O_y$ are simple objects (they only have attributes and no sub-objects). $\text{Sim}_j(O_x, O_y)$ defines the similarity rate between $O_x$ and $O_y$. It computes the weights ($w_i$) of the degree of similarity of their p respective attributes. Each attribute is weighted ($w_i$) according to its importance in the object. (1) computes the local similarity between objects $O_x$ and $O_y$. The signs are sub-objects of syndrome objects, which are sub-objects of clinical picture objects, which are components of disease objects. This composition hierarchy provides k new similarity functions which are the average of local similarities of the n sub-object components (2). The similarities have to be considered at each composition level using the zoom effect to detect local sub-object similarities.

$$\text{Sim} : C \times C \rightarrow [0,1] \quad \text{Sim}_j(O_x, O_y) = \frac{\sum_{j=1}^{n} \text{Sim}_j(O_x, O_y)}{n} \quad (2)$$

$\text{Sim}(O_x, O_y)$ represents the similarity of two composed objects $O_x$ and $O_y$ of a classe C. n is the number of sub-objects. We define $d : C \times C \rightarrow [0,1] : d(O_x, O_y) = 1 - \text{Sim}(O_x, O_y)$ (by definition). $O_x$ is the unique element of A, thus iff $O_x = O_y$, $d(O_x, O_y) = 0$ and $\text{Sim}(O_x, O_y) = \text{Sim}(O_x, O_y)$ (by definition). Consequently, $d(O_x) = \text{Sim}(O_x, O_y)$ is the membership function of $O_x$ and thus, the operators $\{ \cap, \cup \}$ can be used. The evaluation of disease objects relies on the fuzzy membership function $\text{Sim}$. It is necessary to model the evolution of objects to be able to do comparisons.

4.3. Time and analogy of object behaviour
We focus on defining operators to compare objects’ evolution at different granularity levels during a period of time. The Case Based Reasoning approach (CBR) needs analogical reasoning modes.

$$\text{Sim}_{t_x, t_y}(O_x, O_y) = \sum_{t_x, t_y} \text{Sim}_i(O_x, O_y) \quad (3)$$

$p$ is specified as an indivisible amount of time. The smaller the value of $p$, the more accurate is $\text{Sim}_i$. (3) defines the fuzzy temporal membership function $\text{Sim}_i$ in the interval $[t_x, t_y]$. System objects are observed at periodicity $p$. Thus $\forall i, t_i = t_{i+1} + p$. Therefore, the time interval $[t_x, t_y]$ defines the number $(t_y - t_x)/p$ of observation periods, where $i$ is the integer division.

4.3. Implementation of CBR relying on an OO KBS
Most of the present O-O languages are compiled. This constraint impedes real-time object prototyping during knowledge base design. Python and Java can dynamically load new modified classes, and dynamically specify and modify objects according to knowledge expressed by the expert and formalized by the designer. Thus, the tool flexibility favors a quick design of KBS prototypes in complex domains like health and management, where time is essential in decision making.

**Function editor and Function interpreter**
In order to provide prototyping facilities the designer must be able to create, modify, test and delete internal object functions dynamically during the design session.

**Application objects**
Application objects contain internal instructions that provide access to object knowledge base queries and carry out object state evaluation according to the KBS goals, the states of knowledge objects, and user queries and answers. The analogical reasoning mode is implemented by a CBR approach used in medical ethics (Colloc and Léry, 1997).

**Figure 3.** The Case Based Reasoning Cycle

The CBR cycle (Figure 3) memorizes and retrieves experience used in solving similar previous problems. This capability allows CBR to express previous experience knowledge and thus to use it to enhance the decision in new similar situations. The basis of CBR is that, by computing distances (Dubitsky et al., 1997) between new cases and previously stored, indexed cases, and retrieving those cases that are similar, we can apply the corresponding decision and actions to solve this new problem, expecting that what was good one time, will be good several times (Gupta, 1994).

**CONCLUSION**
We have presented the classical triple: model, method and computer-aided tools. We have also presented an example of reasoning mode implementation using a CBR approach with analogy based on a fuzzy temporal membership function. This function is used in the CBR cycles to index and retrieve cases. A class library is currently implemented in JAVA. We are now trying to provide automatic operators to dynamically modify and load knowledge objects based on this approach. Thus, our future work involves refining the method and providing a complete library with metaprogramming features.
REFERENCES


BIBLIOGRAPHIE

JOEL COLLOC holds a Ph.D. (1990) in computer sciences by the National Institute of Applied Sciences (INSA) of Lyon and a MD (1985) by the medical faculty of the Claude Bernard University (Lyon 1) France.
He is currently professor in computer sciences in the Le Havre University and was previously associate professor in the Jean Moulin Lyon 3 University. His main research topics are DSS and knowledge bases in medicine, CBR and multi-agent systems, nervous system modelling and cognitive sciences. He is a member of the LEACM laboratory ISH, University Lyon 2.

PETER SUMMONS was a Systems Engineer in industry.
He has a Graduate Diploma in Education. Since 1986 he lectured at The University of Newcastle Australia, where currently he is a member of the Faculty of Science and IT.
He holds a Ph.D in computer sciences by the University of Newcastle, Australia.
His main research interests are health informatics, DSS and system design.
A SIMULATION-BASED EXPERT SYSTEM FOR PROCESS DIAGNOSIS

Yusong Pang, Hans P. M. Veeke and Gabriel Lodewijks
Delft University of Technology
Mekelweg 2, 2628 CD Delft, the Netherlands
E-mail: Y.Pang@WbMT.TU/Delft.NL

KEYWORDS
Simulation, Expert Systems, Knowledge Acquisition, Case-based Reasoning, Intelligent Diagnosis

ABSTRACT

Expert systems have been widely applied to a variety of domains including the field of intelligent industrial diagnosis. However, knowledge acquisition has been often considered as the bottleneck in expert system development. This paper describes the architecture of an expert system in industrial process diagnosis, including simulation-based knowledge generation, fuzzy knowledge representation and case-based diagnostic reasoning. In this paper, a methodology of acquiring knowledge and building up knowledge bases based on software simulation is introduced. The efficiency and accuracy of the simulation-based methodology had been verified by experimental results of performing such an expert system in a hydraulic brake system.

INTRODUCTION

Fault detection and diagnosis have been considerably interested in recent years because of the increasing requirements for automation, efficiency, reliability and safety in industrial processes. On-line monitoring and diagnosis of operational performances and conditions are important for plant safety and process maintenance profit. In reality, tasks of process diagnosis in responding abnormal events are according to the knowledge, the experience, and the mental and physical status of process operators.

Automated fault diagnosis systems can help operators to make fast and accurate decisions under abnormal conditions. Intelligent diagnostic methodologies offer solutions of technical problems by automating the diagnostic procedure and improving the performance of industrial processes. Nowadays artificial intelligence (AI) technologies have been matured enough to preserve domain knowledge in order to use the past successful experiences for decision-making in the future. Expert systems (ES) have been widely used in domains where human experts are not available or the cost of inquiring an expert is high. Applications of ES have covered fields from medical diagnosis, chemical analysis, geological exploration, computer configuration, plant operation, marine navigation, to real-time process diagnosis and control [Liao 2005].

However, the knowledge acquisition problem has been commonly considered as a major bottleneck in the development of expert systems [Wu 2003] although a wide variety of ES has been built. In many industrial systems, the knowledge is not available for the synthesis of a diagnosis system. All diagnosis methods assume firstly the existence of particular knowledge for instance the existence of a numerical database corresponding to the various operating modes of the process, or the existence of experts able to verbalize their experience of a given process, etc. [Toscano 2002].

Practically, four knowledge sources of expertise, field measurement, process simulation, and enterprise information systems (e.g. enterprise resource planning (ERP) and manufacturing execution system (MES) [Tao 2004]) enable the performance of the knowledge acquisition for an ES in industrial fields. However, even the best domain specialists do not have complete experience for building efficient knowledge bases since the complexity of industrial processes; measuring data and gathering operational knowledge from industrial application fields and enterprise information systems will probably take several years before sufficient knowledge is collected; and some operational conditions are never allowed to actually appear in field measurement. To reduce the required development time and effort for an ES, it is possible to build up the knowledge base with knowledge generated by software model and its simulation.

This paper presents a simulation-based diagnostic ES for industrial processes. One purpose of this research is to investigate the possibilities of using industrial simulation to generate desired knowledge for intelligent diagnostic reasoning. After a brief overview of the introduced diagnostic ES in this paper, three stages, the simulation-based knowledge generation, the fuzzy knowledge representation and the case-based diagnostic reasoning, of building such an ES are introduced. The simulation-based ES had been validated by experimental results.

OVERVIEW OF THE DIAGNOSTIC EXPERT SYSTEM

ES, as a branch of AI, has been developed in last decades. The basic idea of ES is that the expertise may be transferred from human specialists to computer. The knowledge, or the expertise, is stored in computer and then is retrieved for specific advices or solutions when needed based on the inference of computer. In this situation, an ES acts as a consultant which gives advices and explanations and potential logic. Thus applications of
ES are critical in decision-making and problem solving in fields of industry process monitoring, detection and diagnosis.

In the diagnostic ES introduced in this paper, the processing and analyzing of information are automated to fully explore the possibilities of anticipatory control and predictive maintenance. Knowledge of operational conditions, operational actions and maintenance strategies can be built up in a database system – the knowledge base. The monitored operational condition can then be assessed by algorithms combined with the built knowledge bases, and the proper operational actions can be recommended or automatically carried out. The basic concept of the ES is shown in Figure 1.

![Diagram of ES Concept](image)

**Figure 1. Basic concept of the diagnostic ES**

The application field supplies facts or other information to the ES and receives expert advice or expertise in response. Internally, the ES consists of two main components. The knowledge base contains the knowledge with which the reasoning engine draws conclusion. These conclusions are ES’s responses to the application field queries for expertise [Giarratano 1998].

The structure of the introduced diagnostic ES in this paper is shown in Figure 2.

![Diagram of ES Structure](image)

**Figure 2. Structure of the diagnostic ES**

This system is constructed in three stages.

In the knowledge generation stage, the knowledge of supporting process diagnosis is derived from simulation. Process model simulates and explores the effects of process operational conditions. Quantitative and qualitative forms of knowledge are abstracted from results of simulation. Generated knowledge can be readily represented and stored into underlying knowledge base to support diagnosis reasoning.

In the knowledge representation stage, the knowledge derived from simulation are represented as cases with ES recognizable forms and stored into knowledge base. This stage is accomplished by fuzzy knowledge representation algorithms.

The diagnostic reasoning stage invokes the case-based reasoning mechanism. Cases derived from simulation contain the simulated process situation description, its relevant process discovery and known solution. For newly monitored process situations, the process discovery and the solution are those desired knowledge which will be retrieved by an non-modeled associative case completion algorithms in this stage.

**KNOWLEDGE GENERATION**

Classical fault detection and diagnosis methods based on limit value checking of some important measurable parameters do not simulate deep process activity. One of the advantages of modeling techniques that comes from classical numerical fault detection methods is the possibility of detecting developing faults at an early stage (fault prediction). [Angeli 2001]. Nowadays the attempt to model-based diagnostics and control of industrial processes is well-known [Korbicz et al, 2004]. Many such industrial applications are reported where complex industrial installations consisting of numerous pieces of equipment and other hardware, controlled in the open loop by human operators who apply their own knowledge and skill acquired during long-term activities [Moczulski 2004].

In order to build up desired knowledge base through the simulation of industrial process, a process model will be developed and implemented. The resulting software model is used to generate knowledge for the diagnostic ES.

To reach the generation of knowledge, the software model is able to simulate accurately enough the real system in both healthy operational conditions and failure modes. The objectives for the development of the model are to gather knowledge about the behavior of the real system during both normal operation and operation after the introduction of failures, and to collect the gathered knowledge in a database that can be used by the linked ES to determine the occurring failures in the real system.

For simulation and the comparison of important parametric values from a simulated system and the corresponding measured values form the system performance, there are some requirements for the overall structure and functional design of the model. Firstly the model should be based on process physics and contain sufficient adjustable parameters to enable matching between measured and simulated results; secondly the model should be dynamic since most process activities under continuously changing operating conditions; thirdly the model should have standardized input and output parameters and supply measurable parameters for inputs and outputs; fourthly the model should be modular because the process is possible to replace and add components.

**KNOWLEDGE REPRESENTATION**

The specific output of the simulation and the original output of field measurement are not suitable to further reasoning process of a diagnostic ES. The task of knowledge representation stage is to represent simulated and monitored operational conditions as cases which can be used for intelligent reasoning.

Due to the abundance, complexity and uncertainty of industrial knowledge, knowledge representation is an especially difficult and time-consuming task. Most
knowledge sources or actual instances in real-world applications contain fuzzy or ambiguous information. Expressions of the domain knowledge using fuzzy descriptions are thus seen more and more frequently. For many knowledge-intensive applications, it is important to develop an environment that permits flexible modeling and fuzzy querying of complex data and knowledge including uncertainty. [Koyuncu, et al, 2005]. In industry, it is hard and unnecessary to give exact definitions or descriptions of industrial concepts and relationships among concepts. As what a process operator does, it is also not necessary to use precise information for understanding industrial events. To express vagueness and imprecision of industrial events and their relationships the theory of fuzzy logic is employed in knowledge representation.

To overcome limitations of traditional way to acquire knowledge from a industrial process, the knowledge representation stage is based on aims of identifying the operational situations of the process from the data, then creating a situation description to describe events of process, and finally, integrating this knowledge in the knowledge base (the case base). Three steps and three concepts are involved in this stage:

The data-handling step: The historical database covers representative periods. Data from each time period were considered as a particular data set. Great influences exist in the data set such like abnormal values, missing values or strange values. A statistical analysis was carried out to filter some signals and to remove the redundancy of the database. Then the final database was then structured in the format which contained the time period in rows and the variables in columns.

The classification step and concepts of attribute and event: Once the historical database was structured in a rows and columns format, it was fed to a clustering tool in order to conduct the classification step. Each cluster should represent a group of variances within a certain time period characterized by a particular situation of the process. Figure 3 shows examples of such a clustering which are able to represent most of possible variance patterns of industrial events.

![Cluster Diagram](image)

Figure 3. Clustering of industrial events

Variance quantity ($A_q$), variance pattern ($A_p$) and variance response level ($A_r$) are three fuzzified attributes for describing any industrial variance. For any monitored parameter, $A_q$ represents the quantification of the variance which could be high, middle or low; $A_p$ represents the pattern of the trend of the variance which could be smooth in/decrease, vibrated in/decrease or fly-up/down, or jump up/down; and $A_r$ represents the intensity of the variance which could be acute/dull, fast/slow, or medium. Attributes represent how a particular variance.

An event is defined as the combination of three attributes which denote details of the variance of the event. An event can be represented:

$$E_n (A_{q_n}, A_{p_n}, A_{r_n})$$

The case representation step and the concept of process situation description: The decision-making of the ES is performed by case-based reasoning. Cases stored in case base are considered as old cases and named as completed cases. Since the initial setting of simulation condition is known and the knowledge of simulated condition is derived from simulation, the content of a complete case is made up of the process situation description, known operational discoveries and its relevant solutions. In contrast, newly monitored operational situation is considered as new case and represented as incomplete case which contains only the process situation description. A situation description ($S_n$) is a description of a specific operational condition. Any situation can be described by the combination of its relative events:

$$S_m (E_{m1}, E_{m2}, \ldots, E_{mn})$$

Situation description provides the information of how current situation related parameters vary. Situation description is the main part of case representation. With adding the known discovery and solution which had been preset during simulation, a complete case is represented as:

$Case_n :: S_m (E_{m1}, E_{m2}, \ldots, E_{mn}, Discovery, Solution)$

DIAGNOSTIC REASONING

The main task of diagnostic reasoning is to retrieve operational discoveries and solutions from old cases stored in case base when a new incomplete case comes. Instead of rule-based reasoning and other inference technologies in traditional ES applications, the case completion process uses an algorithm of non-modeled association which is a type of content-addressable memory (CAM). It includes common technique such as hash table for speeding-up the knowledge retrieval process. The idea of associative memory device was originally introduced by Vannevar Bush in 1945 [Bush 1945] which is a technique of using association memory to overcome the computational complexity and the complexity of knowledge management. The non-modeled
association utilizes an associative lookup as opposed to an indexed lookup. Associative lookup is inherently very flexible based on the ability to perform similarity and proximity based lookup as opposed to the more brittle and explicit characteristics of index-based lookup because it can be understood as co-occurrence inference which avoids the complexity of modeling and knowledge retrieval. The principle of non-modeled associative case completion is shown in Figure 4.

<table>
<thead>
<tr>
<th>Case</th>
<th>Situation Description (S)</th>
<th>Goal</th>
</tr>
</thead>
<tbody>
<tr>
<td>$E_1$</td>
<td>$A_1, A_2, A_3, A_4, A_5$</td>
<td>$E_2$</td>
</tr>
<tr>
<td>$E_2$</td>
<td>$A_6, A_7, A_8, A_9, A_{10}$</td>
<td>$E_3$</td>
</tr>
<tr>
<td>$E_3$</td>
<td>$A_{11}, A_{12}, A_{13}, A_{14}$</td>
<td>$E_4$</td>
</tr>
</tbody>
</table>

Figure 4. Principle of associative case completion

In Figure 4, attributes in white color present in both complete and incomplete cases; attributes in gray color present only in complete case but not in incomplete case; goal fields denote the missing knowledge which new case is looking for. Only the attributes present in the incomplete case are used in associative computation. Results are determined based on the frequency of co-occurrence of attributes within the goal field. The association algorithm completes some partial or incomplete information into a recognizable condition. It is applied by evaluating the similarity between new and past operational situations and then to retrieve the lacked knowledge of incomplete cases such like operational discoveries and maintenance solutions from past complete case. Case completion algorithm is also one solution of the missing-value problem of field measurement.

SYSTEM PERFORMANCE AND EXPERIMENTAL RESULTS

The diagnostic ES and its software model are validated against field measurements. Validation involves the evaluation of the output accurateness, the system stabilities and statistical analysis based on theoretical process knowledge. A test facility of hydraulic brake system had been applied for evaluating the model, the simulation and the ES.

The Test Facility

The analyzed brake system consists of three components: a hydraulic disc brake, a belonging hydraulic power unit and a controller. The controller monitors the speed of the brake disc and depending on this speed, it controls the amount of oil the hydraulic power unit supplies to the brake. This delivered amount of oil determines the braking force with which the brake is applied to the brake disc and as a result the deceleration of the brake disc.

The facility of field test is shown in Figure 5. The motor has a nominal speed of 1500rpm and drives the low speed shaft through a poly-V-belt transmission with a ratio of (3.21:1). Normally the low speed shaft goes into a gearbox where the speed is increased and transmitted to the high speed shaft.

Figure 5. The test facility of hydraulic brake system

The Simulink Model

A dynamic model of the hydraulic brake system as it functions within the test facility has been developed and implemented. The mathematical software model has been built in software package Simulink. The model is able to generate simulated operational data and to gather information about the process values that are the indicators for the most important failure causes of the hydraulic brake system. Simulation results are sent to the ES and represented as complete cases stored into knowledge base.

Four main parts of the brake system, the hydraulic disc brake, hydraulic power unit, the controller and the brake disc, are included in the model. For the aim of modularity of the model, which results in a better overview and maintainability of the software model, the four physical parts of the system each forms a separate subsystem. The relations of subsystems and their causalities had been known before the systems were modeled. Four parts of the system and their relative parameters are shown in Figure 6.

Figure 6. The Simulink model

Where

$v_{- pv}$ is the velocity of brake disc;
$x\%$ is the pulse width modulation of controller;
$x$ is the displacement of the brake pad;
$dx$ is the velocity of the brake pad;
$p_3$ is the pressure in brake cylinder;

$F_{\text{brake}}$ is the braking force;

$T_{\text{max disc}}$ is the temperature of the brake disc.

Figure 7. Example of model matching

Verification is the initial evaluation of the model mainly based on theoretical process knowledge. Verification should show the ability of the model to describe the physical process. Matching is understood to be the adjustment of parameters in the model, such that the simulated outputs approximate the measured data as accurately as possible over the entire operational range. To reach the purpose of model verification and matching, steady state tests and dynamic tests had been executed. Figure 7 shows an example of results from one of dynamic tests.

Actually the verification and matching of the model had been quite easily done since the trends of process values generated by the model corresponded quite well with the trends in the measured data from test facility.

**User Interface**

The ES gives output with the code of newly monitored situation, the code of retrieved situation, the indication of possible system failure mode and relative operation solution, and the confidence level of case retrieval. The situation code is the simplified process situation description. It is sequentially listed attribute representations of all relevant events. Indication of retrieved solution originally comes from its simulation initial setting. The confidence level denotes how similar two cases are. Figure 8 shows an example of the output of the ES when the failure mode of grease on brake disc occurs.

![User interface of the ES](image)

Figure 8. User interface of the ES

**Results**

The simulation-based ES had been tests by effects of failure modes in the hydraulic brake system on the behaviors of both the simulated and measured process values. For the current software model seventeen failure modes that can be both simulated and measured from test field are most interested.

The ES can be evaluated based on the output it gives. During field tests, the operational condition of the measured data offered to the ES is known, which means that the failure mode is known. Therefore, it is easy to verify if the retrieved process discovery and decision-making solution given as output by the ES corresponds with the failure mode as measured. Figure 9 shows the evaluation results of the ES during the blocks of ten seconds braking time, offering the measurement during failure mode of “control pressure low”.

From experimental results it can be concluded that the simulation-based diagnostic ES gives correct enough maintenance decisions.

![Evaluation of ES output](image)

Figure 9. Evaluation of ES output

**CONCLUSIONS**

It is becoming increasingly clear that knowledge systems are playing a major role in the modern industrial process diagnosis. One unique aspect of the introduce diagnostic ES is that the solution of bottleneck problem of knowledge acquisition during developing an ES is
achieved by the use of simulation in discovering failure causalities and possible operational solution in industrial processes. The methodology of simulation-based knowledge generation shows its efficiency and accuracy of building up knowledge base for an intelligent system. Fuzzy knowledge representation enables system to easily represent outputs of simulation for diagnostic reasoning. Based on results of experimental implementation, it is concluded that the simulation-based diagnostic ES provides accurate enough outputs for process diagnosis and decision-making.

REFERENCES:


Bush, Vannevar, 1945, As We May Think, The Atlantic Monthly.


MULTI SKILLED MAINTENANCE
OF A HIGH SPEED RAILWAY

Michiel de Gijl(1), Jaap Ottjes(2), Jaap Hagestein(1), Gabriel Lodewijks(2)
(1) Infraspeed Maintenance bv. e-mail: Michiel.De.Gijl@infraspeed.nl
(2) TU Delft e-mail: j.a.ottjes@tudelft.nl

KEYWORDS
Railway, maintenance, process-oriented simulation, multi-skill.

ABSTRACT
Periodic inspection and maintenance of railway lines is performed by maintenance teams. This paper concerns a preliminary investigation of the influence of using multi skilled teams on maintenance costs. A simulation model to determine the relationship between skill composition of teams and the maintenance costs has been prepared and applied on the case of a high speed railway line that is under construction. All maintenance tasks were given but preliminary periodic maintenance schedules had to be determined. The quality of the maintenance has been expressed as the percentage of maintenance tasks that are completed in time. It is concluded that a balanced composition of skills per team may reduce the cost of maintenance significantly.

INTRODUCTION
The high speed railway connection (HSL) from Amsterdam to the Dutch-Belgian border is currently under construction. It will connect the western part of the Netherlands, the ‘Randstad’, with the high speed European railway network. The HSL track is 125 kilometres long. 85 kilometres of the route consists of newly laid high-speed track. The maximum speed will be 300 km/h. The line will be completed in 2007. Figure 1 shows the HSL track with branches to the cities of The Hague and Breda.

The consortium Infraspeed is responsible for the design, construction, financing and maintenance of the HSL rail systems. Infraspeed is building the superstructure (the railway system) of the HSL. For the 25-year period following completion of the superstructure a contract comprises the management and maintenance of the entire line under the authority of the rail manager ProRail. In the contract with the government, Infraspeed guarantees 99.46% availability of the line during 25 years. During this contract period, the Dutch government will pay Infraspeed an annual fee for making the HSL infrastructure available. The sum paid depends on the actual availability: if Infraspeed fails to realise the required availability, the fee is reduced (HSL Zuid web site 2006).

Figure 1 HSL track (HSL Zuid web site 2006)

In this paper the focus is on preventive inspection and maintenance. In particular the influence of the number of competences (skills) in maintenance teams on maintenance costs has been investigated. In the next section some maintenance aspects are discussed and the research question will be formulated. After that the simulation model of the preventive maintenance processes is explained and the results of the simulation experiments are presented. Finally conclusions are drawn.
MAINTENANCE

The purpose of maintenance of the HSL is to guarantee a predefined safety level and prescribed line-availability. The maintenance processes are currently under development. In this section the maintenance policy will be discussed and a way to evaluate the performance of the maintenance system is proposed.

Maintenance policy

Maintenance will include periodic inspection and preventive maintenance, corrective maintenance and reactive maintenance.

- Periodic inspections and preventive maintenance are defined as those activities required for determining and evaluating the actual condition of the asset and for maintaining the desired condition and operating standard.
- Corrective maintenance is defined as unplanned activities that can be carried out during a regular maintenance shift and/or possession.
- Reactive Maintenance is defined as unplanned activities that must be carried out to immediately restore track availability even if there is no possession planned. Reactive maintenance may lead to non-availability which will cause penalties.

In this paper periodic inspection and preventive maintenance are considered on the basis of pre-established intervals of time and employing pre-printed checklists (TCRP 2003).

All preventive inspection and maintenance activities to be carried out have been specified and are available in maintenance manuals. The activities have been subdivided into tasks. Each task has its specific periodic maintenance interval, work to be done and skill needed for the operation. The preventive maintenance tasks will be automatically generated based on a 'preventive maintenance schedule'. The initial length of the maintenance interval depends on the assessment of the deterioration of the asset in the time interval and the increase of risk expected. The periodic maintenance intervals will be fine-tuned during the actual operation to obtain the economic optimum balance between system availability and maintenance costs. The preventive maintenance schedule is not yet available however. In this work maintenance schedules are required; consequently preliminary simplified maintenance schedules have to be generated.

The HSL

The HSL is being built out of a number of sub systems including the rails, the electric system (overhead wires and transformer stations), the system for communications, safety and signalling, the sound barriers, balustrades and fences, the facilities in the tunnel buildings, and emergency facilities and ventilation systems in tunnels. For safety reasons maintenance actions are not allowed near the track during train operation. To that end the area where the trains are running is defined by the so called ‘train operating envelop’ that is shown in Figure 2.

Maintenance in the train operating envelop is only allowed during a restricted (5 hours) time period during the night, when no trains are running. During the rest of the twenty-four hours period maintenance activities are only allowed on objects outside the operating envelop. During the day the work of the following night is prepared.

Research question

Traditionally, maintenance teams are single skilled. The main goal of this work is to verify the supposition that the use of multi skilled team compositions may lead to a reduction of maintenance teams needed and thus to reduction of costs. In order to compare different skill compositions it is necessary to have a measure for the quality of the maintenance.

A maintenance performance measure

In the public transport sector the term “quality of service” (QoS) is used for the evaluation of a transport service, (Tervo et al, 2000), (ERTMS/GSM-R, 2003). The QoS is composed of a number of performance measures and even has qualitative components, (TCRP 2003). Both line availability and safety are related to the QoS. In this work a performance measure for the quality of maintenance is defined using the following reasoning. Preventive maintenance task intervals are determined on the basis of the risk assessment in the case that the maintenance fails to occur. It can be argued that the risk of line unavailability increases if a preventive maintenance task is delayed. The main reason of delay will be the unavailability of maintenance teams. The more teams available the less delays will occur, but the higher the cost will be. Consequently there is a trade-off between the costs and the quality of the maintenance. In this work the percentage of preventive maintenance activities that are on time are defined as a simplified measure of the quality of maintenance, it will referred to as “maintenance performance”. Because of this simplified definition of the maintenance performance, the values measured do not have an absolute meaning in terms of risk of failures. It will only be used to compare the quality of a number of maintenance team compositions.

MODELLING

In order to evaluate the research question, the maintenance system is simulated. Simulation has been chosen because it leaves the possibility to extend the model with corrective and reactive maintenance activities and all related stochastic aspects in a later stage. The “process interaction method” is used as a modelling method (Zeigler, 2000), (Fishman, 2001). The method can be characterized by identifying the system element classes and describing the sequence of actions of each
one. The sequence of actions of an element is called its process. In a process, so called “time consuming” commands appear. Examples are: work, wait, drive suspend and standby. A class is further characterized by its attributes. A special class is the “Set” class. A Set may contain system elements and is very useful to define data structures. In practice this method boils down to the decomposition of the system into relevant classes of elements, preferably patterned on the real world elements of the system, the specification of the attributes and the description of the processes of the active classes. Process interaction modelling has a near resemblance to object oriented modelling. For process interaction simulation an appropriate simulation tool is needed. The first programming language applying process interaction is “Simula” (Birtwisle, 1973); two recent tools are Silk (Healy and Kilgore, 1997) and Tomas (Veeke and Otjes, 2000 and 2002). Tomas, that is based on Delphi has been applied in this project. Next the main element classes are discussed.

Element classes and attributes

Class HSL
- WorkZoneSET Set with all Work Zone’s

Class WorkZone
- TravelDist Travel distance between WZ and Yard
- Location

- NofTeams Number of Teams present in WZ

Class Task
- name
- StartTime day from where the Preventive schedule will start
- Interval periodic interval
- Required Shift shift in which the Task can be performed
- Required Skill Skill required
- Duration time needed to execute the task
- Amount of men # of men required
- WZ work zone where the Task is done
- Nr of Duplicates number of same Tasks in the Work zone
- PROCESS Process of a Task

Class Team
- Name Team name
- NofMen # of men in a Team
- SkillSet Skills available in the team
- Hourrate Cost per hour per Team member
- Shift Shift in which the Team has to work
- Location The location where the Team is stationed
- PROCESS Process of a Team

The structure of the model is shown in Figure 3.

The operational working is determined by the processes of the elements. The processes of a maintenance team and a maintenance task will be shown in pseudo code.

Process of a Team
A team is only allowed to work in a regular shift. If the task to be done is within the train operating envelop only the night shift is appropriate. The selection of a task is rule based. Delayed tasks always have priority.

Figure 2 the train operating envelope
Figure 3. Structure of the model.

Figure 4 Screen shot of the simulation model
Process of a Task
A Task first creates a copy of itself and starts the process of that copy with a delay of “interval” time units. In this way the repetition of the Task is guaranteed. Next the Task enters the proper PMTaskSet and waits to be executed by a team. In a steady state situation the executing will be finished before the repetition interval elapses. If the Task is not executed at the end of the interval, it leaves the PMTaskSet of its WorkZone and enters the DelayedTaskSet of its Workzone and continues waiting for execution. ‘Now’ is the current system time. The number of delayed tasks is used to calculate the performance quality.

Schedule generation
At the moment of this investigation no preventive maintenance schedule was available. Therefore the model is used in a slightly modified version to generate a preliminary schedule and to tune the workforce. To do so all tasks are released at the start of the simulation run: Now = 0. The process of a Task however only starts at the very moment that it is executed for the first time. In this way the work load is leveled out. If there are not enough teams available the work in progress will increase continuously and if there is a surplus of teams there will be underutilization. These effects are used to tune the workforce needed. Figure 4 shows a screen shot of the simulation model with a plot of the numbers of tasks in the system and the under utilization of shifts.

EXPERIMENTS AND RESULTS
In this section the model is applied to the HSL case. Table 1 shows the subsystems of the HSL. Only the predefined maintenance tasks of the first five sub systems have been used. A number of basic scenarios have been tested, applying different skill compositions of teams. First preliminary preventive maintenance schedules have been derived for each scenario and after that the number of teams was fine-tuned to evaluate the benefits of teams with more than one skill. All runs cover a period of 10 years.

<table>
<thead>
<tr>
<th>Table 1 subsystems of the HSL and skill required</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. civil structures, the substructure (CIV):</td>
</tr>
<tr>
<td>2. noise and fencing (NFE):</td>
</tr>
<tr>
<td>3. signalling (SIG):</td>
</tr>
<tr>
<td>4. traction power distribution (TPD):</td>
</tr>
<tr>
<td>5. track (TRA)</td>
</tr>
<tr>
<td>6. ancillary electrical and mechanical equipment (AEM):</td>
</tr>
<tr>
<td>7. buildings and miscellaneous (BMI):</td>
</tr>
<tr>
<td>8. communications (COM):</td>
</tr>
</tbody>
</table>

Generating preliminary schedules
In order to generate a schedule, all maintenance tasks have been released at the start of the simulation run. The tasks are executed successively by the available teams. To get a stable situation, the number of teams is varied as well as the number of teams allowed in one work zone. Figure 5 shows a situation that does not stabilize. The number of issued tasks is growing steadily.

Figure 5 Tuning the workforce: growing number of Tasks waiting for execution.

In Figure 6 the result is shown of tuning both the number of teams available and the number of teams allowed in one work zone. The situation is in balance and the underutilization of the teams is minimal. The output of the run provides a preventive maintenance schedule consisting of a list of all preventive maintenance tasks and corresponding starting dates. Table 2 shows a part of a maintenance schedule, consisting of a sequence of predefined maintenance tasks.

Figure 6 Tuning the workforce: Stable execution of the Tasks.
The preventive maintenance schedules have been used for the further simulation experiments to establish the maintenance performance. In stead of releasing all tasks at a time, they are released on their planned starting times. Figure 7 shows the results for single skilled teams with a maximum of 2 teams per work zone. The workload has levelled out. The peaks in the numbers of released tasks can be explained by the fact that a large number of short tasks have been clustered.

![Figure 7 Released tasks as a result of preventive maintenance schedule over a period of 10 years.](image)

The hypothesis in this work is that using multi skilled teams will reduce labour costs while keeping the maintenance performance at a desired level. The next step is simulating all skill compositions and, in case of underutilization, decreasing the number of teams. Table 3 shows a part of an input file with team definitions. The skill name refers to the sub system of Table 1.

<table>
<thead>
<tr>
<th>Table 3 Part of team definition input</th>
</tr>
</thead>
<tbody>
<tr>
<td>{ HSL Team input }</td>
</tr>
<tr>
<td>13 [Number of Teams]</td>
</tr>
<tr>
<td>TeamTRAS1 [TName] {6}</td>
</tr>
<tr>
<td>3 [NroMen]</td>
</tr>
<tr>
<td>1 [Efficiency]</td>
</tr>
<tr>
<td>90 [HourRate]</td>
</tr>
<tr>
<td>Night [Shift]</td>
</tr>
<tr>
<td>South [Location]</td>
</tr>
<tr>
<td>2 [NroSkill]</td>
</tr>
<tr>
<td>TRA [Skill 1]</td>
</tr>
<tr>
<td>SIG [Skill 4]</td>
</tr>
<tr>
<td>TeamNFES [TName] {9}</td>
</tr>
<tr>
<td>3 [NroMen]</td>
</tr>
<tr>
<td>1 [Efficiency]</td>
</tr>
<tr>
<td>90 [HourRate]</td>
</tr>
<tr>
<td>Night [Shift]</td>
</tr>
<tr>
<td>South [Location]</td>
</tr>
<tr>
<td>1 [NroSkill]</td>
</tr>
<tr>
<td>4TV [Skill 2]</td>
</tr>
</tbody>
</table>

The final results of the experiments are shown in Figure 8, in which the maintenance performance and the maintenance costs have been plotted as a function of the team compositions. Scenario 1 (Sc1) shows the initial result with single skilled teams. Scenario 2 (Sc2) has been obtained by combining at most two skills per team. The logical combinations of skill couples have been prepared by Infraspeed as a part of an earlier test, concerning only two skills per team. The results of the scenarios Sc3 -Sc8 are obtained by further varying skill composition starting from scenario 2 and tuning the number of teams. For each scenario the maintenance performance has been determined and compared with the single skill case. Only results with a maintenance performance at least as high as measured for the single skilled case, are accepted. In Figure 8 for each scenario the maintenance performance and the matching costs are shown. It is concluded that several multi skill compositions perform at lower costs at least as good as the single skill case. It was decided to proceed with the further development of scenario 8 (Sc8) in which at most 3 shifts are combined.
CONCLUSIONS AND FURTHER RESEARCH

A simulation model has been developed and implemented to determine both the maintenance costs and maintenance performance as a function of the skill composition of the maintenance teams. All required preventive task definitions were available at the start of the simulation project. All preliminary maintenance schedules however had to be generated. A modified version of the model was used for that purpose.

It is concluded that using multi skilled maintenance teams and tuning the composition of skills in the teams reduces the number of teams required. The maintenance costs can be reduced significantly compared with only single skilled operation. Aspects that not have been taking into consideration are the extra costs of training personal to obtain multiple skills and to employ them.

Further research is necessary to incorporate stochastic aspects and corrective and reactive maintenance and to optimize the tuning of skills and scheduling the teams.

REFERENCES


HSL zaid web site 2006 : http://www.hslzuid.nl/hsl/uk/Organization/index.jsp


Figure 8 Labour cost and maintenance performance versus maintenance team skill composition
LOGISTICS
SIMULATION
Towards a best strategy in inventory decision making and demand forecasting for intermittent demand

Katrien Ramaekers and Gerrit K. Janssens
Operations Management and Logistics
Hasselt University
Agoralaan - Building D, 3590 Diepenbeek, Belgium
e-mail: {katrien.ramaekers,gerrit.janssens}@uhasselt.be

Keywords: intermittent demand, forecasting, inventory management, simulation, optimisation

Abstract

Demand forecasting is one of the most crucial aspects of inventory management. For intermittent demand, i.e. demand peaks follow several periods of zero or low demands, forecasting is difficult. A simulation model is used to study a single-product inventory system facing demand of the intermittent type. In this paper, a research approach is described to find a best strategy in combining inventory decision making and demand forecasting for intermittent demand.

1 Introduction

Inventory systems have to cope with uncertainty in demand. The inventory control literature mostly makes use of the Normal or Gamma distributions for describing the demand in the lead-time. The Poisson distribution has been found to provide a reasonable fit when demand is very low (only a few pieces per year). Less attention has been paid to irregular demand. This type of demand is characterised by a high level of variability, but may be also of the intermittent type, i.e. demand peaks follow several periods of zero or low demands.

Demand forecasting is one of the most crucial aspects of inventory management [9]. However, for intermittent demand, forecasting is difficult, and errors in prediction may be costly in terms of obsolescent stock or unmet demand [8]. The standard forecasting method for intermittent demand items is considered to be Croston's method [2]. This method builds estimates taking into account both demand size and the interval between demand occurrences. Despite the theoretical superiority of such an estimation procedure, empirical evidence suggests modest gains in performance when compared with simpler forecasting techniques [7]. Furthermore, the choice of the forecasting method can have an impact on the inventory management policy that is best used.

Preliminary research demonstrates the presence of an interaction between demand forecasting and inventory decision making for intermittent demand using a simulation model to study a single-product inventory system facing demand of the intermittent type. Therefore, in this paper, a research approach is described to optimise the simulation model in order to obtain the best strategy in combining inventory decision making and demand forecasting. Furthermore, some initial results are discussed.

2 Experimental framework

This section describes the inventory systems and forecasting methods that are used in this research. The study focuses on a single-product inventory system facing demand of the intermittent type. To generate intermittent demand, demand occurrence and demand size are separately generated. The demand occurrence is generated according to a first-order Markov process with transition matrix
\[ P = \begin{pmatrix} p_{00} & p_{01} \\ p_{10} & p_{11} \end{pmatrix}, \]

where \( p_{00} \) is the probability of no order in the next period when there has been no order in this period and \( p_{10} \) is the probability of no order in the next period when there has been an order in the current period. Individual order sizes are generated using a Gamma distribution with shape parameter \( \gamma \) and scale parameter \( \beta \).

The simulation run length is set to 52 periods and 10 replications are made for each simulation run.

The total costs and the performance (number of stock-out periods and number of stock-out units) of the inventory system are determined.

### 2.1 Inventory systems

There are two general types of inventory systems: continuous review models and periodic review models. In continuous review models, the stock level is always known whereas in periodic review models, the stock level is determined only every \( R \) time units.

In this research, two periodic review models are used. The first one is the \((R, s, S)\) system. This means that every \( R \) units of time, the inventory level is checked. If it is at or below the reorder point \( s \), a sufficient quantity is ordered to raise it to \( S \). The second system \((R, s, Q)\) is similar to the \((R, s, S)\) system but uses a fixed order quantity \( Q \) instead of an order-up-to-level \( S \).

A deterministic lead-time \( L \) is assumed. Three possible review periods are considered: review period equal to lead-time, review period equal to twice the lead-time and review period equal to half the lead-time.

The following costs are considered: unit holding cost per period \( C_h \), ordering cost \( C_o \) and unit shortage cost per period \( C_s \). The simulation starts with an initial inventory level \( I_0 \).

### 2.2 Forecasting methods

The standard forecasting method for intermittent demand items is considered to be Croston's method. However, in practice, single exponential smoothing and simple moving averages are often used to deal with intermittent demand.

**Exponential smoothing** is probably the most used of all forecasting techniques. The single exponential smoothing (SES) method is easy to apply, the forecast is calculated as:

\[ F_t = \alpha X_{t-1} + (1 - \alpha) F_{t-1} \quad (1) \]

where \( \alpha \) is the smoothing constant that determines the weight given to the most recent past observations and therefore controls the rate of smoothing or averaging. It is commonly constrained to be in the range of zero to one.

The assumption of the **moving average** forecasting method is that a future value will equal an average of past values. The number of past values used to calculate the forecast can vary. The four-period moving average forecast is calculated as:

\[ F_t = W_4 X_{t-4} + W_3 X_{t-3} + W_2 X_{t-2} + W_1 X_{t-1}. \quad (2) \]

**Croston's method** [2] was developed to provide a more accurate forecast of the mean demand per period. Croston's method applies exponential smoothing separately to the intervals between nonzero demands and their sizes. Let \( I_t \) be the smoothed estimate of the mean interval between nonzero demands, and let \( S_t \) be the smoothed estimate of the mean size of a nonzero demand. Let \( q \) be the time interval since the last nonzero demand. Croston's method works as follows: if \( X_t = 0 \) then

\[ S_t = S_{t-1}; I_t = I_{t-1}; q = q + 1 \quad (3) \]

else

\[ S_t = \alpha X_t + (1 - \alpha) S_{t-1}; I_t = \alpha q + (1 - \alpha) I_{t-1}; q = 1. \quad (4) \]

where \( \alpha \) is the smoothing parameter. Combining the estimates of size and interval provides the forecast:

\[ F_t = S_t / I_t. \quad (5) \]
2.3 Experimental design

The experimental design includes three qualitative factors: the forecasting method, the inventory management policy and the review period. In addition, depending on the choice of the qualitative factors, a set of quantitative factors are part of the experimental design. If the (R, s, Q) inventory management policy is used, the safety stock SS and order quantity Q are the parameters to optimise. If the (R, s, S) inventory management policy is used, the safety stock SS and order-up-to-level S are the optimising parameters. For single exponential smoothing and Croston’s method, the smoothing parameter α is optimised and for moving averages, the weights of the past values are optimised.

This research aims to decide on the optimal combination of forecasting method, inventory management policy and review period. Furthermore, the optimal settings for the safety stock, the fixed order quantity or order-up-to-level and the parameter(s) of the forecasting method are determined.

3 Approach

Because of the dependence of the quantitative factors on the choice of the qualitative factors, we use the research approach described in this section.

For every combination of forecasting method, inventory management policy and review period, the optimal values of the quantitative factors are determined. This is done using two different optimisation methods: Taguchi method and tabu search. These two methods are shortly described below. Once the optimal values are found, the best combination of forecasting method, inventory management policy and review period is chosen.

3.1 Design of experiments: Taguchi’s method

Design of Experiment (DOE) Techniques [5] provide a way to set up the complete experimental design before the experimentation process begins. The experimental points are chosen in order to cover the search space as completely as possible. Design of experiment methods can in general only be applied to discrete variables, so the first step before applying a DOE-method consists of choosing a limited number of discrete values in the domain of each continuous variable.

Several schemes for setting up experimental designs are known from literature. The Taguchi design is an interesting technique. The first step is to rank the n relevant values of each decision variable and give them a level number from 1 to n. The next stage is to set up the experiments. This is usually done using specially constructed orthogonal arrays containing a number of rows. Each row defines one experiment to be carried out with the corresponding levels for the variables.

Three discrete values are chosen in the domain of each of the three quantitative, continuous factors. These values are shown in Table 1, where SS, S and Q are calculated using the following formulas:

\[ SS = z\sigma_{R+L} \]  

where \( \sigma_{R+L} \) is the standard deviation of demand over the review period and the lead-time. The value \( z \) depends on the desired service level.

The fixed order quantity \( Q \) is determined using the formula of the Economic Order Quantity EOQ:

\[ Q = \sqrt{\frac{2\bar{X}C_o}{C_h}}. \]  

The order-up-to-level \( S \) is the sum of the safety stock and the average demand over the vulnerable period:

\[ S = SS + \bar{X}(R + L). \]  

3.2 Tabu search

Tabu search uses a local or neighbourhood search procedure to iteratively move from one solution to the next: in the neighbourhood of the first, until some stopping criterion has been satisfied. To explore regions in the search space that would be left unexplored by the local search procedure and escape local optimality, tabu search modifies the neighbourhood structure of each solution as the search progresses. The solutions admitted to the
new neighbourhood are determined through the use of special memory structures. Tabu search uses both long-term and short-term memory, and each type of memory has its own special strategies [3, 4].

Tabu search is a heuristic optimisation technique developed specifically for combinatorial problems. Very few works deal with the application to the global minimization of functions depending on continuous variables. The method we propose in this paper is based on [1, 6]. The purpose in these papers is to keep as close as possible to original tabu search. Two issues must be examined: the generation of current solution neighbours and the elaboration of the tabu list.

To define a neighbourhood of the current solution, a set of hyperrectangles is used for the partition of the current solution neighbourhood. The $k$ neighbours of the current solution are obtained by selecting one point at random inside each hyperrectangular zone.

Once a new current solution is determined, the immediate neighbourhood of the previous solution is added to the tabu list.

As a starting point, the safety stock $SS$, fixed order quantity $Q$ and order-up-to-level $S$ are calculated using the formulas above. A neighbourhood consists of 5 neighbours and the tabu list contains 5 tabu areas. 10 simulation runs are made for each experimental choice. The tabu search is stopped after 200 iterations.

### Table 1: Discrete values of the variables in the Taguchi method

<table>
<thead>
<tr>
<th>Variable</th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\alpha$</td>
<td>0.2</td>
<td>0.5</td>
<td>0.8</td>
</tr>
<tr>
<td>Weight 1</td>
<td>0.1</td>
<td>0.25</td>
<td>0.4</td>
</tr>
<tr>
<td>Weight 2</td>
<td>0.2</td>
<td>0.25</td>
<td>0.3</td>
</tr>
<tr>
<td>Weight 3</td>
<td>0.3</td>
<td>0.25</td>
<td>0.2</td>
</tr>
<tr>
<td>Weight 4</td>
<td>0.4</td>
<td>0.25</td>
<td>0.1</td>
</tr>
<tr>
<td>Safety Stock</td>
<td>SS</td>
<td>SS+2</td>
<td></td>
</tr>
<tr>
<td>Order-up-to-level</td>
<td>S-5</td>
<td>S</td>
<td>S+5</td>
</tr>
<tr>
<td>Fixed Order Quantity</td>
<td>Q-5</td>
<td>Q</td>
<td>Q+5</td>
</tr>
</tbody>
</table>

### 4 Experimental environment

The experimental environment contains the costs of the inventory system and the parameters for generating intermittent demand. The research approach described above, is executed using a single combination of the costs of the inventory system and demand. However, these factors can have an effect on the results that are obtained. An experimental design is set up for these factors and the optimisation phase is repeated for each experimental point.

*Demand occurrence* is generated using a first-order Markov process with transition matrices:

$$ P_1 = \begin{pmatrix} 0.7875 & 0.2125 \\ 0.85 & 0.15 \end{pmatrix} $$

or

$$ P_2 = \begin{pmatrix} 0.5667 & 0.4333 \\ 0.65 & 0.35 \end{pmatrix}. $$

They correspond with a probability of 20% to have demand in a certain period for the first matrix and a probability of 40% to have demand in a period for the second matrix. The *size of demand* is generated using a Gamma distribution with 4 different combinations of the scale parameter $\gamma$ and the shape parameter $\beta$. These values are summarized in Table 2.

### Table 2: Parameters of the Gamma distribution

<table>
<thead>
<tr>
<th>Combination</th>
<th>$\gamma$</th>
<th>$\beta$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>6</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>12</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>4</td>
<td>24</td>
<td>0.5</td>
</tr>
</tbody>
</table>

The levels of the costs of the inventory system are given in Table 3. The initial inventory level $I_0$ equals 5.

This experimental design makes it possible to determine the impact of uncontrollable factors as the cost structure and the demand on the optimal strategy in inventory decision making and demand forecasting for intermittent demand.
Table 3: Levels for the costs of the inventory system

<table>
<thead>
<tr>
<th>Level</th>
<th>( C_a )</th>
<th>( C_h )</th>
<th>( C_s )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>100</td>
<td>2</td>
<td>5</td>
</tr>
<tr>
<td>2</td>
<td>200</td>
<td>4</td>
<td>10</td>
</tr>
</tbody>
</table>

5 Results

The basic configuration of the factors of the experimental environment is set as follows: demand occurrence is generated using a first-order Markov process with transition matrix

\[
P_1 = \begin{pmatrix} 0.7875 & 0.2125 \\ 0.85 & 0.15 \end{pmatrix}.
\]

For the demand size, a gamma distribution with scale parameter 6 and shape parameter 1 is used. The ordering cost equals €100 per order, the unit shortage cost €5 per period and the unit holding cost €2 per period.

When using Taguchi’s method, the optimal solution for this experimental environment is an order-up-to-level inventory management system with a review period equal to twice the lead-time. Croston’s method with \( \alpha = 0.5 \) is best used as forecasting method. The safety stock is equal to 6 units and the order-up-to-level is also equal to 6 units for the optimal solution. This means an order is placed every time the inventory level drops below the safety stock and enough is ordered to raise it again to the size of the safety stock.

When the demand size is doubled, the review period is best set equal to the lead-time. The safety stock and order-up-to-level have higher values but they are still equal to each other. When the demand frequency is doubled, the smoothing constant \( \alpha = 0.8 \) gives the best results. The safety stock is 4 and the order-up-to-level equals 9.

Changes in the costs of the system also cause differences in the conclusions. When the unit holding cost is doubled, \( \alpha \) should be set to 0.8. The order-up-to-level is in this situation best lower than the safety stock. When the unit shortage cost or the ordering cost is doubled, the only difference in the conclusions is the order-up-to-level being smaller than the safety stock.

When Tabu search is used as optimisation method for the quantitative factors, the optimal strategy for the basic configuration of the factors of the experimental environment is an order-up-to-level inventory management policy with a review period equal to the lead-time. Exponential smoothing is the forecasting method that leads to the lowest costs. The smoothing parameter \( \alpha \) does not have significant impact on the results. The order-up-to-level \( S \) is equal to 1 which implies that the safety stock \( SS \) is negative and the reorder point is 0.

When the demand frequency is doubled, the best inventory management policy depends on the cost structure. However, instead of an order-up-to-level \( S \) of 1 unit, the order-up-to-level \( S \) or fixed order quantity \( Q \) is a value between 15 and 30. It can also be noted that when the demand frequency is doubled, Croston’s method becomes less useful as forecasting method. Changing the parameters of the demand size does not have a significant impact on the results.

Changes in the cost structure of the inventory system have a significant impact on the results. If the ordering cost is doubled, it is better to have an order-up-to-level or fixed order quantity that is between 15 and 30, except when the unit holding cost is high and the unit shortage cost is low. When the unit holding cost is doubled, an order-up-to-level of 1 is the best choice, unless the ordering cost and unit shortage cost are also high and the demand frequency is high. Doubling the unit shortage cost leads to an order-up-to-level or fixed order quantity between 15 and 30, except when the unit holding cost is high and the demand frequency is low.

6 Conclusions

Overall, it can be concluded that both optimisation methods lead to roughly similar results but when tabu search is applied, continuous values are used which leads to more accurate results.

The factors of the experimental environment have an impact on the best strategy for combining
inventory decision-making and demand forecasting and there is also interaction between those factors.

For intermittent demand, the best inventory management policy is an order-up-to-level policy with an order-up-to-level $S$ equal to 1. The reorder point equals 0 and the safety stock is negative. The best forecasting method depends on the cost structure. The parameters of the forecasting method do not influence the results significantly.

References


Biography

Katrien Ramaekers graduated as Master of Business Economics, option Technology at the Limburg University Centre in 2002. In October 2002, she started as a Ph.D.-student at Hasselt University. Her main research interest is on the integration of simulation and optimisation, especially as a support for complex logistics decision-making and for decision support with limited information in supply and demand. She is a member of the Data Analysis and Modelling research group and of the Transportation Research Institute of Hasselt University.

Gerrit K. Janssens received degrees of M.Sc. in Engineering with Economy from the University of Antwerp (RUCA), Belgium, M.Sc. in Computer Science from the University of Ghent (RUG), Belgium, and Ph.D. from the Free University of Brussels (VUB), Belgium. After some years of work at General Motors Continental, Antwerp, he joined the University of Antwerp until the year 2000. Currently he is Professor of Operations Management and Logistics at Hasselt University within the Faculty of Business Administration. He also holds the CPIM certificate of the American Production and Inventory Control Society (APICS). During the last fifteen years he has repeatedly been visiting faculty member of universities in Thailand, Vietnam, Cambodia and Zimbabwe. His main research interests include the development and application of operations research models in production and distribution logistics.
WAREHOUSE AND MANUFACTURING LOGISTICS DESIGN USING A DATA-DRIVEN GENERIC MODEL GENERATOR

José A. Basto
António C. Brito

Universidade do Porto, Faculdade de Engenharia, Departamento de Engenharia Mecânica e Gestão Industrial
Rua Dr. Roberto Frias; P-4200-465 PORTO; PORTUGAL
E-mail: jbbasto@fe.up.pt, acbrito@fe.up.pt

KEYWORDS
Simulation, Modelling, Warehouse, Manufacturing

ABSTRACT
The nature of warehouse design requires the manipulation of large amounts of data and is often an iterative process that forces the designer to go through the different design phases several times before reaching the final solution. This suggests an integrated computer environment that can give support to the user during all the design phases.

The need for a flexible tool, easier to use, lead to the development of a Decision Support System: AWARD (Advanced WAREhouse Design).

This paper presents further developments of the DSS and shows a successful example of application of the new functionalities: a simulation model of a full automated warehouse developed for EFACEC, a large Portuguese company in the warehouse design, building and consultancy business.

INTRODUCTION
In 1976 the concept of Visual Interactive Simulation (VIS) was introduced by Hurrion (Hurrion 1976). With VIS a image of the model running is displayed on the screen and the user can interrupt the model running, at any time, and interact with it in a way that he can influence the future behaviour of the model.

Visual Interactive Simulation aggregates a set of graphics, visual and interactive techniques which can be incorporated with different levels of extension in a simulation system. In the past years, there have been a large number of situations where VIS was successfully applied to different areas, including warehousing (Dangelmaier and Bachers 1986), (Marin et al. 1998), (Feliz-Teixeira and Brito 1999) and (Burnett and LeBaron 2001).

Physical locations and travel distances are crucial in warehouse modelling. The warehouse layout is a key factor in the success of the system. When developing warehouse models the spatial factor is essential for a good representation of the real system. A warehousing simulation system using the Data-driven Generic Model approach should then have the following characteristics:
- a user friendly interface using warehouse terminology;
- an effective way of defining the physical layout in order to enable the model of movement with the required level of accuracy;
- simulation elements which allow a correct modelling of warehouse logic.

These needs lead to AWARD (Advanced WAREhouse Design), a Decision Support System for warehouse design and management. The model uses the SIMVIS libraries and utilities described in (Bastos and Moreira da Silva 1985).

This simulation library was written at the Management Group (GEIN) of the Mechanical Engineering and Industrial Management Department (DEMEGI) of the Faculty of Engineering - Porto University - Portugal. The programming language used was FORTRAN and later C++. The library is based on the simulation event approach. AWARD development has been user oriented and incorporates the experience and background discussed above, as described in (Brito and Basto 1992).

Although SIMVIS was used to build simulation models, its main objective was the development of Data-Driven Generic Models. The SIMVIS author's strategy was to conceive a system with the basic tools necessary to support the development of model generators for specific domains. With the Data-Driven Generic Models the user can build models with a friendly interface using a dialogue that he can understand and without a great need of simulation expertise. A set of support libraries were also developed, at GEIN, to allow the software to be easily ported to other operating systems or be adapted to work with different graphics terminals. This was achieved by building separate libraries for the low level I/O and graphics modules, that are device and/or operating system dependent, and for the high level I/O and simulation modules, that are device and operating system independent. Each simulation model is defined by programming the events and all the user interaction and graphics display animation.
The model structure is composed of three modules. The configuration module (CONFIG) allows the definition of the coordinates, the number of aisles and the number of racking levels. In the IDUMP module the user is asked to define the start date and time and the duration of the simulation. Next it generates the internal simulation model structure. The simulation module allows the running of any previously defined configuration.

AWARD DSS EXTENSIONS

Grouping and Marshaling

The need to extend the array of the models generated by AWARD DSS to simulate the grouping of deliveries to the warehouse and the marshaling for expedition orders, lead to extend functionalities to the DSS.

In the case of the operation of grouping, full pallets accumulated in the corresponding zones. For marshaling, pallets accumulated there may be simple (single product) or mixed (boxes of different products, picked for a specific customer). The goal of these operations is to diminish the global load (or unload) times of the trailers, what is obtained at a cost: the reduction of the useful area of the warehouse and the increase of the number of handlings. The rewards achieved by the extra operation of marshaling are particularly relevant in the case of warehouses designed for the supply of products in units of boxes, not full pallets. In this scenario, being able to pick-up and marshaling these boxes before the arrival of the expedition truck is an important step to increase the volume flow of expedition bay and the utilization rate of the truck.

The grouping of deliveries allows the reduction of the unloading times of the trucks that dock to the reception bays and enable the execution of parallel tasks that frequently occur, such as the verification of incoming supplies and its initial quality control.

We may create more than one zone of the above cited types, allowing to model diverse situations that include grouping and marshaling zones with different physical locations or different types of racks.

The new information concerning the grouping of incoming pallets must be added to the products data file. One of its fields indicates the name of the grouping intermediate storage zone. If this data field is blank, none of the products of this family will stop at this zone. Otherwise, the products that belong to the specified family stay in the grouping zone during a minimum holding time that is also defined in the same file. The data format for the operation of marshaling is similar to the one just described for the grouping storage.

With the new operations of grouping and marshaling, the transfer mechanisms truck - warehouse and warehouse - truck are now more complex. Processes that where scheduled and done in a single moment, must now be considered in two phases, involving careful synchronization and verification of conditions of space availability.

The role of entity PALLET, acquired a new importance, because it is now responsible for the trigger of intermediate events, as for example, “ENTER IN GROUPING”. Entity PALLET is updated during its traveling in the warehouse, in particular the attribute that identifies the operation in which this entity is involved and the attributes that keep the pointers for origin and destination in each phase of the transfer.

Boxes picking for new type of “mixed pallets” add a bigger information load for the transfer process, in which the trips to the racks with picking cells can be as many as the boxes that are part of this special pallet. We opted to create an additional entity, MIXED PALLET, to deal exclusively with this new process.

Pick-up and delivery transfer stations

To increase the efficiency of a warehouse, we may use transporters who have the capacity to operate in aisles of reduced width (narrow-aisle storage/retrieval truck or automated storage/retrieval system) and racks of much bigger height (denser storage) than those that are possible to operate using conventional forklift trucks.

These types of transporters suffer, however, of an important limitation: in the generality of the cases, its operation is limited to the aisle of access to the racks it is dedicated. Due to this restriction, it became necessary to create interface points (from now on named TRANSFER STATIONS) between this constrained equipment and conventional material handling equipment with flexible access to the warehouse.

This new resource is basically a special type of rack that functions as a "buffer", where all pallets that are directed to or come from a rack served by a dedicated transporter must stop. Not representing useful volume of storage, the stations must be as small as possible, what leads to an increase of the complexity of the mechanisms that deal with the problem of pallets position reserve and “locking”, the scheduling of events to unblock tasks dependent on the availability of free positions in stations and the proper transference between different stations.

We choose the storage rack that will have pick-up and delivery stations, eliminating the hypothesis of it being already a station. Then the racks that will work as transfer stations are pointed, allowing that the same rack assumes the two roles. The relation storage rack - transfer station is univocal: one given rack can only have one station of each type, while one station may serve multiple storage racks.
This correspondence between racks led to an increase of the structure of data “racks”, at the CONFIG program, where the routines of creation, elimination or repetition of racks have now to contemplate the new transfer stations.

The IDUMP program was expanded with the creation of the new entity TRANSFER STATION, with new queues of free and busy cells (for exit and/or entrance, depending on the characteristics of each station) and the extension of the set of attributes of the rack to keep the pointers for its eventual transfer stations. The possible transfers inside the warehouse are presented in Figure 1.

In the beginning of the process of handling a pallet to its position of destination, is verified if the “target” rack has assigned a delivery station. If this condition applies, a sub-routine gets a pointer to a free position in the transfer “buffer”, being this cell immediately locked for the transit of entity PALLET that triggered the event. In the absence of a free position, the PALLET is blocked until a cell in the station is freed.

The transfer of pallets through exit stations involves problems related with the existence of free positions in the marshaling zone (in the eventuality of the routing includes this zone): it does not justify the locking of a position in an pick-up station if it is forecasted that this resource (in general, scarce) will be blocked until becoming vacant a needed downstream position. Therefore, it was decided to never reserve a cell at stations without before-hand make sure that all the remaining passage points are available.

This priority of keeping the stations as available as possible was also included in the new logic of the restocking task of “boxes picking points”. If the trip includes a short stop at the exit station of the storage rack (full pallets) and another stop at the station of entrance to the rack of boxes picking, this task will only reserve position in the stations when it will be possible to lock simultaneously the cells in the two midway buffers.

The task of boxes picking raises the biggest problem in the reserve of position in input/output stations, because this case may be much more than a single trip from a rack to a truck (or vice versa), always in the same flow direction. The order of boxes may have multiple delivery points, located in dispersed racks of the warehouse, what may lead to the situation that is represented in Figure 2, in which the single cell stations serve both functions (this is a worst case scenario).

The procedure to perform includes the following steps:
(i) In the arrival of a pallet to the transfer station I (see Figure 2), the pointer for the delivery cell of boxes “1” is obtained from the picking list; Next, it verifies if the rack to which this delivery point belongs is assigned (or not) an input transfer station. In the affirmative case, it is verified if it exists a free position in this station. If this condition is also met, we will be in conditions to initiate next check (before trip “A” starts);
(ii) Still in the transfer station I, verifies if the station II is also an exit station. If affirmative, the task of “boxes picking” immediately reserves the same position in station II as exit and entry point, anticipating the return of pallet to the station II (trip “B”, in figure 2)
The underlying motive to this procedure is to prevent that a mixed pallet is blocked in an aisle. The subdivision of the task “boxes picking” still compels numerous tests of accessibility of the transporters to the various points where the routing takes the pallet, in contrast to the procedure followed by the first version of system AWARD, in which the same transporter were in charge to fulfill the order “boxes picking” from start to finish.

**Manufacturing Logistics and AS/RS**

EFACEC entered a new bid for building an automatic warehouse designed to supply raw material to machines integrated in a large production system. These functionalities required new options to be added to the simulator AWARD, not yet prepared to model manufacturing logistics systems with Automated Storage and Retrieval Systems (AS/RS).

On input data side, it was necessary to create a new data structure to describe the operations of each machine. The new files (one for each machine) include the information of the operations to be executed in the machine, namely: the codes of the necessary raw materials, the amounts to be used; set-up and processing times for each operation.

It was also needed to widen the logic of the warehouse zones, in order to define zones of direct supplying to machines and buffers for empty pallets.

Given that the EFACEC is specialized in the conception and production of automatic warehouses, aiming dense storage in warehouses and distribution centres, it became imperative to increase the range of types of available transporters in order to include the AUTOMATIC TRANSPORTERS (robots destined to the transport of pallets from/to the racks). By technical restrictions (in general this equipment moves on a rail track, being held in place by a guide in form of “T” on the top track), this AS/RS is always constrained to linear movements, keeping its orientation during all the simulation.

This new transporter graphical representation reflects its special shape and the double platform, if it is equipped to perform a dual cycle on a single trip (for example, picking up an empty pallet from a machine supply bay and deliver a full one). The coding of this logic proved crucial to achieve the operational level needed to manage hard industrial problems, like the one presented as an example in the next section. Finally, the multiple new events to model this production environment were created in the generator IDUMP.

The production machines to be supplied originated the new entity MACHINE. The attributes of this entity keep the information for its functioning, as pointers for the supplying stations, raw materials to be used, timing of the events related to the operations, etc, keeping also the statistics of the times in each one of its possible states, indicated at the icon of the machine by a small colored flag:

(i) SETUP: Preparation of the machine (blue flag)
(ii) OPERATION: Processing of the materials in the machine (green flag);
(iii) STARVING: the machine stops by lack of raw material (red flag);
(iv) REQUEST: the machine send to the central controller a supplying order (yellow flag). REQUEST may coexist with state (SETUP) or with state (OPERATION), preceding always state (STARVING)

**INDUSTRIAL APLICATION EXAMPLE**

The new functionalities of system AWARD were used for the simulation of an automatic warehouse designed by EFACEC for a component supplier in the competitive automotive industry.

The problem required the supply of raw materials to 38 machines. The layout is presented in Figure 3, a screen shoot of the full model, in which we can see the storage racks, with seven levels, in two sections, each served by an automatic transporter.

![Figure 3: Initial Situation of Manufacturing Layout](image)

The raw material pallets arrive in the factory by a receiving bay located to the left of the racks, where also are located the zones of accumulation of empty pallets.

The supplying of the machines is done through the cells of the first level of the racks, having each machine three supply stations. A small “translation” software program was prepared for importing data from a master production plan (supplied from real data by the final customer of EFACEC). It creates the list of shop orders for each machine, in accordance with the file structure defined in the AWARD system.

The main measurement to get from the simulation is the share of the time at STARVING state of the machines, as well as statistics of utilization of transporters and material flow.

**Results**

The studied scenarios were based at the above detailed physical implantation, modelled to scale of the actual
layout, using as control parameters the initial fill level of the warehouse and the dispatching priorities assigned to the automatic transporters, that where fitted with the “double platform” to permit dual cycle after preliminary runs indicated material flow problems.

It was decided to test a “worst case scenario” to obtain a “lower bound” result. Thus, the simulated warehouse had just a minimum reserve stock and its operation priority was given to the raw material reception.

In this boundary situation, where it is necessary to supply all the 38 machines almost simultaneously, taking care in the first place of the input of pallets into the warehouse, a few starvation situations occurred during a transient period of about two hours. (See Figure 4, taken 25 minutes after the beginning of the simulation, with a “red flag”). After this provoked crisis fades away, no more supply rupture happens, barely changing final statistics of the simulation.

Figure 4: STARVING Machine being Supplied.

In more normal starting conditions, with an initial fill level of the warehouse as low as 10%, and even keeping the same unreasonable priority of the tasks, not a single machine reached STARVATION status, what demonstrates that the planned material handling equipments had spare capacity to meet the load forecasted for the manufacturing logistics system. Suggestions where made to use SMED methodology to reduce machines setup time, the real hurdle in the factory performance.

CONCLUSIONS

The AWARD model generator and simulation system proved to be a powerful tool to the design and testing of warehouses, enabling managers and engineers that are not experts in the field of simulation to obtain results in a couple of days (instead of months of hard coding by experienced simulators and programmers) to big industrial challenges.

The new described functionalities widen the applicability of system AWARD, adding essential resources to the automatic equipment simulation of materials handling and allowing industrial needs for warehouse zones and transfer stations to be easily included in the models.

The creation of the new logic destined to the problem of manufacturing logistics enabled AWARD to accurately simulate with easy modeling automatic warehouses (with new A5/RS transporters) in production facilities. As a marketing tool for EFACCEC, this model generator with faithful display, proved to be a successful competitive leverage to win commercial bids.

REFERENCES


AUTHOR BIOGRAPHY

JOSÉ A. BASTO, Porto (Portugal), 1963, went to Universidade do Porto, where he studied mechanical engineering and obtained his degree in 1986, staying in the same institution as Lecturer. He obtained his Ph.D. in Industrial Engineering at the Industrial and Systems Engineering Department, Lehigh University (U.S.A.) in 2001. He is Auxiliar Professor at Universidade do Porto since 2001, where he leads research in the field of Flexible Manufacturing. His e-mail address is: jbbasto@fc.up.pt

ANTÓNIO C. BRITO, Porto, 1958, went to Universidade do Porto, where he studied mechanical engineering and obtained his degree in 1981, staying in the same institution as Lecturer where he did his M.S. thesis in 1985. He obtained his Ph.D. in Simulation from the School of Management at the University of Cranfield (U.K.) in 1993. He is Auxiliar Professor at Universidade do Porto since 1993, where he leads research in the field of Visual Interactive Simulation. His e-mail address is: acbrito@fc.up.pt
SUPPLY CHAIN SIMULATION AND OPTIMIZATION
THE DISTINCT INFLUENCE OF VARIOUS CPF FR COMPONENTS ON OVERALL SUPPLY-CHAIN PERFORMANCE

Thomas Thron, Niaz Wassan, Gábor Nagy
Kent Business School, University of Kent
Canterbury, KENT CT2 7PE, UK
Tel.: (+44)1227 827726 Fax.: (+44)1227 761187
Email: T.Thron@kent.ac.uk

KEYWORDS
SCM, Simulation, Collaboration, Distribution, Forecasting

ABSTRACT
In this paper discrete-event simulation is used to point out which particular components of a typical real-world collaborative supply-chain framework promise to be most rewarding. Components evaluated for this purpose are joint promotional activity schedules, seasonality schedules, short/mid-term sales trend forecasts and “intelligent” delivery systems. The analysis uses data from two food-manufacturers and four of their major customers to obtain information about which of the considered CPF modules are worth implementing and to what extent other elements can possibly be neglected for the sake of cost saving, ease of implementation and data protection.

INTRODUCTION
To tackle the matter of supply chain inefficiencies due to a lack of flawless demand information academic research as well as companies’ field studies aim to develop new replenishment approaches which are based on the general idea of information sharing and preferably get along without traditional routine orders. The basic idea is to approach the entire distribution chain as a replenishment-driven process based on reliable forecasts rather than an uninformed carriage system based on rather vague order prediction (VICS, 1998; Simchi-Levi & Kaminski, 2002). The most common and widespread of such automated replenishment approaches are vendor managed inventory (VMI) and collaborative planning forecasting and replenishment (CPF). In a VMI partnership the manufacturer or any other supplier/distributor makes the main replenishment decisions for the consumer. In practical terms the manufacturer monitors the customer’s inventory levels and decides about periodic replenishment actions regarding delivery time and quantity. A particular advantage is the possibility for the manufacturer to recognise and possibly delay non critical replenishments to smooth demand peaks and rank delivery requests according to actual priority (Waller et al., 1999; Kaipia et al., 2002). CPF is an advancement of VMI and describes the sharing of forecast and related business information among business partners in the supply chain to enable automatic product replenishment (VICS, 1998; Ireland et al., 2000). Altogether there is no doubt about the tremendous efficiency gains that can and do result out of a complete change from a ROP based to a CPF delivery system. The most common achievements are higher service levels, lower inventories, fewer and shorter delivery delays or higher fill-loads (VICS, 1999; Hugos, 2003). Nevertheless within this research framework it is often neglected to consider which particular components within a collaborative planning forecasting and replenishment process are worth implementing to achieve the most significant benefits. This would be particularly noteworthy for companies that might want to pick the individual elements out of a CPF framework that are most rewarding for their specific situation whilst avoiding having to take less important components. One of the reasons for doing so can simply be wishing to avoid the costs associated with the introduction, another would be companies’ obvious wish to protect their data and avoid having to share crucial data with other market participants.

In this paper we use discrete-event simulation to investigate which of the chosen components of a CPF system achieve the highest benefits within two typical supply-chain scenarios. The analysis uses data from two food-manufacturers and four of their major customers to obtain information about which of the considered CPF modules are worth implementing within the investigated supply chain framework and to what extent other elements can possibly be neglected for the sake of cost saving, ease of implementation and data protection.

METHODOLOGY
The simulation model that is used in the investigation is based on a business framework and thus real demand and delivery data of an actual medium-size food-company and their four major customers (grocery retailers). We chose to use discrete-event simulation as a well-accepted and somewhat matured methodology of operational research. For a summary of features, advantages and fields of application see Law and Kelton (2000), Pidd (2004) or Brooks and Robinson (2001). Simulation as a time-based modelling tool allows researchers to calculate time-based statistics and just as important, transferable model-code and animation provide an understandable representation of the system acceptable even to non-modellers. Maloni and Benton (1997) recommend using simulation models in particular to critically evaluate possible benefits of supply chain collaboration. One of the main advantages within that framework is the ability to evaluate interdependencies among random effects that may cause a serious degradation in performance even though average performance characteristics of a system appear to be reasonable (Shapiro, 2001). For that particular reason, discrete event simulation has most successfully been used to study flexibility in manufacturing systems (Gupta and Goyal,
The simulation model that served the following analysis was specifically designed to evaluate possible benefits that arise from initiation of information-sharing between manufacturers and retail companies, thus using ongoing demand and inventory data from a varying number of customers to improve production and delivery scheduling. Some of the involved companies have recently undertaken a collaborative replenishment pilot study based on unveiling detailed demand and inventory data on a weekly basis on store level as well as on distribution centre level. Experience and data gained throughout this pilot study proved to be very valuable to validate the simulation model outcome for the particular customers and thus the entire simulation framework. In addition to the above, further underlying data like weekly sales quantities, promotional activity schedules, promotional impact estimations, seasonal factor data, production scheduling, market trend/market share analyses or delivery strategies have been obtained in close cooperation with the manufacturers and additional support by the involved retail companies. The applied simulation model has been designed in a way to reflect the market conditions of the particular companies. Although the model features an evident amount of customisation, it should still be general and flexible enough to be representative for a wide variety of typical supply chain frameworks and market conditions of various enterprises of similar size and delivery structure. A graphical representation is shown in Figure 1.

The Simulation Model

The simulation model used for the investigation consists of three main suppliers serving the particular manufacturer which is then serving the distribution centres (DCs) of four major customers. Moreover, these DCs serve several retail outlets each. Both particular companies under investigation do business with more than four customers. Nevertheless, the four involved retail-companies are their key clients and account for about 80% of total turnover in case of manufacturer 1 and about 85% for manufacturer 2. Thus additional influence of the remaining customers should be negligible due to their minor size. The general model framework is used for both companies nevertheless investigations are run individually since a vast amount of adjustments are needed and underlying data has to be changed to appropriately model each individual manufacturer’s business structure.

Figure 1: Outline of the modelled Supply Chain (one CPFR-customer scenario)

Even though it is part of the simulation model, the investigation will not discuss any issues arising out of the relationship between the particular manufacturer and its suppliers of raw material. Data that is used to run this part of the system is rather taken as a given. Within the distribution framework between manufacturer and retailers - deliveries for each particular customer are either scheduled via a traditional reorder policy (ROP) or follow a CPFR approach. The ROP case simulates a scenario where order requests are triggered by the retailer according to inventory level dropping below a particular lower bound without prior knowledge of the manufacturer regarding point of time or quantity. Within this scenario the manufacturer has to rely only on the standard demand as to say average sales throughout the year with some seasonal adjustments as to face e.g. higher demand in summer and lower in winter as it is the case for the investigated product group of manufacturer 1. Production scheduling thus has to depend on past experience and a certain amount of estimation and does not involve recent sales information.

The CPFR/VMI approach simulates a scenario where sales data on store level as well as inventory levels at retail outlets and distribution centres are available to the manufacturer on a weekly updated basis. Furthermore collaborative effort makes it possible to obtain precise seasonal sales deviation factors as well as underlying long and mid-term market-trends. Another important component of a CPFR system is the collaborative setup of detailed promotion schedules combined with a promotion-impact estimation. Based on this market intelligence a sales forecast is generated and thus implemented in the model that serves the CPFR part as basis to determine production levels and estimate replenishment points and necessary quantities. The forecast is generated for each customer individually taking historic demand, price changes, promotional activity schedules and impact level estimations, weekly seasonal factors and long term market trends into account. These factors are then taken to decompose recent sales data supplied by various retail outlets which again serve to generate a short term trend forecast based on regression and exponential smoothing techniques. The final demand estimation is then obtained through reintegration of the underlying factors into the generated short term trend forecast. This forecasting procedure represents the concrete practice within one of the involved companies. The actual replenishment is commonly arranged between manufacturer and retail-companies via a VMI approach. Thus the manufacturer supervises the inventory level of the retail-company’s distribution centres and replenishment requests are triggered by DC-inventories reaching a particular reorder point that is determined by the above mentioned sales forecast. This system leads to a high level of demand transparency and makes it thus possible to deliver a scarce product to the location that it truly needs most urgent. The retail outlets in the simulation model are always connected with their DC via a CPFR/VMI replenishment system. The delivery is on a daily request basis with average replenishment intervals approximately once a week. This can be considered as common practice for many kinds of products and within most major retail companies (VICS, 1999; Holmström et al., 2002). Thus the replenishment strategy between DC and outlets will not be
further analysed. Within this outlined framework the main focus of the following analysis will be on how individual components of a typical CPFR replenishment system can increase efficiency of the investigated supply chain performance measures.

Some further background information

The demand simulation part of the model is based on more than three years of actual sales data that have been obtained from four particular retailers for each of the two companies under investigation. The sales data under consideration is characterised by a certain volatility in demand behaviour, remarkable seasonality and particularly intensive promotional activities. The two manufacturing companies that apart from the involved retailers provided most of the data for this research are medium sized enterprises with annual turnovers of 75 and 160 million Euro respectively. Both produce grocery items within very distinct food categories. They each manufacture a variety of similar kinds of products in various forms and shapes and additionally in various sizes. The products that were chosen for the analysis were taken out of sets of data that contained detailed sales information about the top five selling SKUs of the particular manufacturers. Altogether sales behaviour from all products within a particular group was found to be almost identical. Replenishments are commonly made ordering the entire set of products in similar amounts and any kind of sales promotion always affects an entire set. Thus the analysis focuses on particular SKUs instead of a whole set since several test runs with each of the other four products within a set resulted in a virtually identical outcome.

Both companies use weekly production cycles which adjust production to expected demand within the following four weeks. Production is capacity constrained as overtime work can only be applied to a very limited amount due to available facilities and union agreements and can commonly only be set up at the beginning of the week. Due to stringent capacity restrictions the inventory will be set on a fairly lean level as this goes along with the companies’ long term inventory policy that average stock-level of each inventory class A SKU throughout the year must not exceed 1.5 (company 1) or 1 (company 2) average weeks’ demand. This is due to the extensive assortment of products, limited durability and somewhat insufficient, non-expandable inventory space. Thus, the inventory limitation is set to hold for all investigated cases. Due to these restrictions the investigation will assume fixed inventory levels and investigate changes in service level figures instead which is different from most previous approaches. The simulation software used was Simul8 Professional R12 (Simul8 Corp, 2005). The model is prior to each run set up with sufficient startup inventories to be able to fulfill replenishment and incoming orders respectively. To further reduce most of the initialization-bias and assure an unflawed observation result-collection is initiated after a 13 weeks warm-up period. Due to a substantial degree of incorporated variation regarding inter-arrival/processing distribution timings, each setting is run 10 times to achieve sufficient confidence levels for all output variables of interest.

Identification of several typical CPFR components

To be able to address the research subject it is preliminarily necessary to clearly identify, isolate and later reintegrate particular CPFR components in the supply chain model. After investigating the practical aspects of a CPFR system implementation on a theoretical as well as at a practical level (i.e. considering the companies’ circumstances), four components could be isolated that should be somewhat representative for a typical collaborative replenishment system: promotional activities, seasonal sales deviations, short/mid term sales-trend forecast and an “intelligent” delivery system. The promotional activities component basically incorporates the collaborative effort between the manufacturer and retail customers to work out detailed promotional schedules and impact estimation. Thus particularly the manufacturer can adjust its production and delivery schedule to be able to cope with temporary demand boosts. To be able to incorporate this into the model past promotional schedules were obtained for the involved products with the help of the involved companies and an impact analysis has been carried out to judge about typical sales deviations that are caused by promotions. Seasonal factors as a further component tackle sales deviations that show regular pattern within each year. The importance of seasonal factor analysis depends on how strong these sales deviations are. The two product groups under investigation have moderate (Company 2) to strong (Company 1) seasonality. Thus a detailed investigation should be highly recommended. The use of 3 years of detailed sales data and further 5 years of historic figures, together with personal judgement of management, made it possible to obtain very comprehensive seasonal factors on a weekly basis. Prior to CPFR implementation (ROP state) company 1 handled seasonal sales deviations in a way that for three winter month production was usually decreased by 20% whilst during summer it was increased by 20% (altogether summer production is thus 50% higher than during winter). Company 2 used to set production up by 10% throughout winter whilst reducing it by 10% during summer.

The short-term trend forecast incorporates a sales forecasting system that uses exponential smoothing and regression techniques to identify short/mid-term sales deviations caused by some sort of abnormal behaviour. For example sales deviations due to unusual weather conditions are an issue since both product groups have a certain weather dependency for part of the demand. Other factors of influence are varying vacation periods, food trends (dietary-waves etc.) or competitors’ promotions. Other short term influences could be due to delivery problems, strikes, product expiration, price or competition related. The variety of influences makes it already obvious that it is not easy to distinguish between events that need to be included into production planning and delivery scheduling and others that should not. There has to be a certain balance between long-term smoothing and incorporation of short term deviations which the parameter settings of the particular forecasting method needs to take care of (for further details about the implemented forecasting approach see Thron, 2004).
The “intelligent” delivery system is typically an integral part of a CPFR system since it enables to prioritize deliveries according to actual urgency. Therefore it uses available information about inventory levels at DC centres and retail outlets to decide how a scarce product should be distributed best to avoid service level gaps. The ROP case in this analysis uses a random (impartial) approach to prioritize simultaneously incoming orders. Often companies tend to have an internal prioritization setting that prefers major customers from smaller ones. This case and general influence of various prioritization strategies have been investigated in a previous paper (Thron et. al., 2005).

It has to be mentioned here that even though we will analyse supply chain settings that include all sorts of independent combinations of the above factors there is some sort of dependency between them. Seasonal factors as well as promotional activities are the only two factors which do not rely on any other component. The trend forecast component for example actually uses promotional activity data as well as seasonal factors to decompose the sales data prior to computation. This is very much necessary since without deseasonalisation and “depromotionising” a short term sales trend forecast would be pretty useless. Altogether resulting influences on the entire system outcome should be minimal since the actual production planning does in this case not incorporate seasonal factors or promotional activity data. The same applies for the intelligent delivery approach. Since a decision about how to distribute scarce products can only reasonably be made on the basis of a demand prediction, the implemented forecasting system is used to make these decisions. Once again this does not involve production planning, hence all figures for these individual settings will be sufficiently valid we just have to keep in mind that for example implementing an “intelligent” delivery system could in real life not be done without initially obtaining detailed information about promotional activities, seasonal factors etc.

**Collaboration requirements for identified CPFR components**

The four above identified collaboration modules can all be considered as reasonably important for an advanced distribution process but require nonetheless very different levels of collaboration and information exchange between the various supply chain members.

Seasonal factors are certainly the most unproblematic component since they can be obtained by a simple one-time data analysis of several years of historic sales information. Thus data protection issues and IT utilisation cost should be minimal. Certainly more effort is necessary to incorporate promotional activities since this necessitates at least annual meetings to layout promotional activity schedules and intense collaboration to obtain realistic promotional impact estimates. Partners would also have to incorporate some sort of emergency scheme that informs about possible alterations regarding time and impact. The highest requirements regarding time, effort and cost would however be necessary for sales trend forecast and “intelligent” delivery system. Both would require not just seasonal factor and promotional activity analyses but also up to date information about inventory levels at store level. This is necessary to generate a reasonable sales forecast which is an essential requirement for both of these collaboration modules. Additionally a sales-trend-forecast would require further investigation about global market trends whilst an “intelligent” delivery system would need up to date information about inventory levels at distribution centre level as well as deeper insight into the distribution process within the particular retail-companies.

**PERFORMANCE MEASURES**

To evaluate the various combinations of the above identified components we use several performance measures that aim to give an impression about the globally achieved supply-chain performance. These performance metrics can be clustered into three groups: Global service level measures taken on store level, delivery accuracy metrics covering supply between manufacturer and retailers’ distribution centres and finally delivery accuracy metrics covering replenishment between retailers’ distribution centres and retail outlets.

**Metrics accounting for the first group**

**Overall Service Level Gap** – this figure accumulates the individual service level gap figures of the four customers and is thus the most important performance measure within the entire analysis. This measure accounts for any occurring gap in supply on store level and thus lost sales and customer/consumer goodwill which has to be seen as the ultimate failure within a supply chain. $S_l \text{ gap}_1/C2/C3/C4$ state the particular figures for each individual customer.

**Median Service Level Gap** – expresses the average gap in supply at store level for just the weeks in which demand cannot be fully met by available inventory. This measure thus gives further insight as to how severe supply-gaps are once they should occur.

**Largest gap in supply** – states the service level gap for the single worst week of supply from any one of the customers.

**Weeks of perfect supply** – states the percentage of weeks within the total investigation timeframe where demand could be fulfilled to 100%. This is taken as an average from individual figures from each of the four customers.

**Production Forecast Accuracy** is meant to give an impression as to what extent increased demand transparency helps improving production forecast accuracy that drives production planning on a four-week time horizon. This metric is also a measure for the quality of the underlying forecast system that uses available promotion schedules, seasonal factors, long and short term market trends and of course recent sales data to provide a reasonably good estimate of forthcoming demand.

**Metrics accounting for the second group**

**Uncritical delays by Manufacturer** – this measure captures the overall percentage of deliveries that were for some reason delayed by the manufacturer and could not be carried out from stock straight away. Nevertheless fulfillment did not exceed the typical order lead time. The delay can be because of unavailability of the particular item or due to certain delivery prioritization policies.

**Critical delays by Manufacturer** – stands for the percentage of deliveries that were postponed by the manufacturer and finally carried out with a substantial delay that made the delivery exceed critical order lead time.
These cases must be seen as rather severe interruptions of order-delivery procedure and are thus the reason for major inconveniences.

**Manufacturer's fill rate** – measures the load level (fill rate) that deliveries obtained on average. Thus it captures the proportion of the initially requested amount that was actually delivered. This measure should optimally be 100%.

**Percentage of perfect deliveries** – captures the proportion of deliveries that were not critically delayed and achieved a fill rate of 100%.

**Metrics accounting for the third group**

**Uncritical delays by Distribution Centres** – this measure captures the overall percentage of deliveries that were delayed by the retail companies’ distribution centres whilst actual fulfilment did not exceed the typical order lead time. The delay can be because of unavailability of the particular item or due to certain delivery prioritization policies.

**Critical delays by Distribution Centres** – stands for the percentage of deliveries that were postponed by the retail companies’ distribution centres and finally carried out with a substantial delay that made the delivery exceed critical order lead time and thus most likely led to a gap in consumer supply.

**Overall DC fill rate** – measures the load level (fill rate) that deliveries from the retail companies’ distribution centres obtained on average. Thus it captures the proportion of the initially requested amount that was actually delivered. This measure should optimally be at a 100% level.

The above measures can be seen as a selection of standard performance measures for the analysis of the considered kind and segment of supply chains and should thus make it possible to obtain a clear picture about the potential benefits from each component combination (Waller et al., 1999; Simchi-Levi and Kaminski, 2002; Hugos, 2003).

**INVESTIGATION OUTCOMES**

Hereafter, we eventually explore what sort of impact each of the four identified CPFR components has on the above defined performance measures. Moreover, we will evaluate every possible combination of these measures to obtain a clearer picture of which components in what combination achieve the highest benefits. To have an initial idea about the situation prior to any CPFR implementation and thus pure ROP as well as the final “full-featured” CPFR scenario we state these two cases initially.

**Table 1: Supply Chain performance for the two benchmark scenarios**

<table>
<thead>
<tr>
<th></th>
<th>Company 1</th>
<th>Company 2</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>ROP</td>
<td>CPF</td>
</tr>
<tr>
<td>Overall SL Gap</td>
<td>5.76%</td>
<td>0.59%</td>
</tr>
<tr>
<td>SL gap Customer 1</td>
<td>4.0%</td>
<td>0.4%</td>
</tr>
<tr>
<td>SL gap Customer 2</td>
<td>9.0%</td>
<td>0.9%</td>
</tr>
<tr>
<td>SL gap Customer 3</td>
<td>5.2%</td>
<td>0.4%</td>
</tr>
<tr>
<td>SL gap Customer 4</td>
<td>7.0%</td>
<td>0.5%</td>
</tr>
<tr>
<td>Median SL gap</td>
<td>13.4%</td>
<td>9.2%</td>
</tr>
<tr>
<td>Largest gap</td>
<td>100.00%</td>
<td>90.00%</td>
</tr>
<tr>
<td>Weeks of p.s.</td>
<td>77.9%</td>
<td>80.8%</td>
</tr>
<tr>
<td>ProFCAccuracy</td>
<td>87.4%</td>
<td>89.6%</td>
</tr>
<tr>
<td>Uncritical delays</td>
<td>43.4%</td>
<td>29.3%</td>
</tr>
<tr>
<td>Critical delays</td>
<td>18.5%</td>
<td>9.2%</td>
</tr>
<tr>
<td>Fill rate Manuf.</td>
<td>19.5%</td>
<td>9.1%</td>
</tr>
<tr>
<td>Perfect delivery</td>
<td>43.3%</td>
<td>58.8%</td>
</tr>
<tr>
<td>Uncritical delays</td>
<td>4.8%</td>
<td>1.0%</td>
</tr>
<tr>
<td>Critical delays</td>
<td>1.5%</td>
<td>0.2%</td>
</tr>
<tr>
<td>Fill rate DCs</td>
<td>98.1%</td>
<td>99.3%</td>
</tr>
</tbody>
</table>

The above table, as well as all those that follow show the particular outcome for Company 1 and Company 2. The investigated performance measures are stated in the same order as they were outlined earlier on.

As we can see from the table, the efficiency improvements of the chosen supply-chain performance measures are enormous comparing the CPFR with the initial settings. Service Level gaps almost vanished whilst critical delivery delay figures went down by 50%. Furthermore the extent of actual supply gaps is far more moderate with median service level gaps being around 3% instead 10-20% before. The situation for the retailers improved as well although figures are not as extraordinary as the improvement between manufacturer and customers’ distribution centres. This is mainly due to replenishment between DCs and retail outlets being already managed by a CPFR system.

Having thus gained a general idea about the level of improvement resulting from a full featured CPFR system, the following analysis will have a look at which individual features or feature combinations achieve the most benefits. Therefore, 14 scenarios were created for each manufacturer, each one featuring a combination of either 1, 2 or 3 of the 4 identified CPFR components. From the particular outcome we will be able to judge which are the most effective factors and factor-combinations and which components do not significantly contribute to an improved overall outcome. Hence we will be able to estimate what particular modules are worth investing in and which others can rather be neglected to save cost and make it unnecessary to reveal too much information to possibly antagonistic market participants.

**One component evaluation**

In the first stage we will have a look at the improvements various performance measures show after one individual CPFR component is introduced. The four components under investigation are illustrated as **PA** - detailed knowledge of promotional activities, **SF** - detailed weekly seasonal factors, **TF** - a trend-prediction according to an underlying forecast-system and **ID** - intelligent delivery procedures that distribute scarce products according to particular demand-urgency. These are compared to the initial pure **ROP** and the full collaboration **CPFR** setup.

**Table 2: Supply Chain performance outcome within one-component evaluation Company 1**

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>ROP</td>
<td>5.76%</td>
<td>4.0%</td>
<td>9.0%</td>
<td>5.2%</td>
<td>7.0%</td>
<td>13.4%</td>
<td>100.0%</td>
<td>77.9%</td>
<td>87.4%</td>
<td>89.6%</td>
</tr>
<tr>
<td>PA</td>
<td>5.52%</td>
<td>3.85%</td>
<td>10.2%</td>
<td>4.2%</td>
<td>5.6%</td>
<td>9.8%</td>
<td>100.0%</td>
<td>77.9%</td>
<td>87.4%</td>
<td>89.6%</td>
</tr>
<tr>
<td>SF</td>
<td>2.11%</td>
<td>1.5%</td>
<td>2.6%</td>
<td>2.1%</td>
<td>1.5%</td>
<td>2.1%</td>
<td>100.0%</td>
<td>77.9%</td>
<td>87.4%</td>
<td>89.6%</td>
</tr>
<tr>
<td>TF</td>
<td>3.78%</td>
<td>3.1%</td>
<td>5.3%</td>
<td>4.2%</td>
<td>6.4%</td>
<td>4.8%</td>
<td>100.0%</td>
<td>77.9%</td>
<td>87.4%</td>
<td>89.6%</td>
</tr>
<tr>
<td>ID</td>
<td>5.76%</td>
<td>4.0%</td>
<td>9.0%</td>
<td>5.2%</td>
<td>7.0%</td>
<td>13.4%</td>
<td>100.0%</td>
<td>77.9%</td>
<td>87.4%</td>
<td>89.6%</td>
</tr>
<tr>
<td>CPF</td>
<td>0.59%</td>
<td>0.4%</td>
<td>0.9%</td>
<td>0.6%</td>
<td>0.4%</td>
<td>0.6%</td>
<td>100.0%</td>
<td>77.9%</td>
<td>87.4%</td>
<td>89.6%</td>
</tr>
</tbody>
</table>

The incorporation of one single CPFR component leads to a noteworthy different outcome for the two investigated companies. In case of Company 1 the use of accurate
weekly seasonal factors for production scheduling seems to be the most rewarding implementation. Integration of only this single factor diminishes the overall service level gap by 3.5%. Critical and uncritical delays decrease significantly. Compared to these achievements the other three components show rather disappointing results. Solely the trend-forecast leads to somewhat noticeable improvements but fails to impress overall. Particularly the incorporation of promotional activities and intelligent delivery system have to be considered ineffective since even though service level gaps diminish, delivery delay figures for manufacturer-DC as well as DC-retailer echelons aggravate.

Table 3: Supply Chain performance outcome within one-component evaluation Company 2

<table>
<thead>
<tr>
<th>Simulation Scenario Company 2</th>
<th>ROP</th>
<th>PA</th>
<th>SF</th>
<th>TF</th>
<th>ID</th>
<th>CPFRI</th>
</tr>
</thead>
<tbody>
<tr>
<td>Overall SL Gap</td>
<td>2.81</td>
<td>1.91</td>
<td>2.59</td>
<td>1.04</td>
<td>2.33</td>
<td>0.63</td>
</tr>
<tr>
<td>SL gap Customer 1</td>
<td>1.8</td>
<td>1.2</td>
<td>1.5</td>
<td>0.6</td>
<td>1.3</td>
<td>0.4</td>
</tr>
<tr>
<td>SL gap Customer 2</td>
<td>4.7</td>
<td>3.1</td>
<td>4.5</td>
<td>1.6</td>
<td>3.8</td>
<td>0.9</td>
</tr>
<tr>
<td>SL gap Customer 3</td>
<td>4.7</td>
<td>3.1</td>
<td>4.3</td>
<td>1.6</td>
<td>3.8</td>
<td>0.9</td>
</tr>
<tr>
<td>SL gap Customer 4</td>
<td>0.6</td>
<td>0.5</td>
<td>0.5</td>
<td>0.5</td>
<td>0.7</td>
<td>0.4</td>
</tr>
<tr>
<td>Median SL gap</td>
<td>9.6</td>
<td>7.3</td>
<td>9.2</td>
<td>5.0</td>
<td>7.7</td>
<td>3.6</td>
</tr>
<tr>
<td>Largest gap</td>
<td>87.1</td>
<td>65.5</td>
<td>76.0</td>
<td>28.6</td>
<td>51.1</td>
<td>20.7</td>
</tr>
<tr>
<td>Weeks of p.s.</td>
<td>52.1</td>
<td>52.7</td>
<td>62.2</td>
<td>65.6</td>
<td>90.2</td>
<td>87.4</td>
</tr>
<tr>
<td>ProdfC Accuracy</td>
<td>92.4</td>
<td>92.5</td>
<td>92.1</td>
<td>94.0</td>
<td>92.4</td>
<td>95.5</td>
</tr>
<tr>
<td>Uncritical delays</td>
<td>44.0</td>
<td>38.6</td>
<td>41.8</td>
<td>32.0</td>
<td>51.5</td>
<td>33.4</td>
</tr>
<tr>
<td>Critical delays</td>
<td>12.6</td>
<td>10.3</td>
<td>12.6</td>
<td>7.3</td>
<td>21.3</td>
<td>6.4</td>
</tr>
<tr>
<td>Fill rate Manuf</td>
<td>95.8</td>
<td>97.9</td>
<td>96.8</td>
<td>93.4</td>
<td>95.8</td>
<td>93.7</td>
</tr>
<tr>
<td>Perfect delivery</td>
<td>34.4</td>
<td>47.7</td>
<td>40.1</td>
<td>49.7</td>
<td>38.4</td>
<td>50.0</td>
</tr>
<tr>
<td>Fill rate DCs</td>
<td>90.5</td>
<td>96.8</td>
<td>98.9</td>
<td>98.7</td>
<td>98.9</td>
<td>99.1</td>
</tr>
</tbody>
</table>

The case of Company 2 reveals a largely different picture. Here the short-term sales trend forecast seems to reveal the highest improvements. Introducing this single factor closes the service level gap by 2.5%. Looking at the other three components makes it seem that detailed knowledge about promotional activities to adjust production planning has a quite remarkably positive effect as well. Service level figures as well as delays improve considerably. However, detailed knowledge about seasonal factors does not seem to have a significant impact on the system compared to Company 1’s outcome which can easily be explained with the fact that seasonal sales deviations are much less significant. On the other hand sales of Company 2 appear to be considerably influenced by short/mid-term market trends which seems to explain the significant improvements in case of introducing a forecasting system that adjusts production scheduling to these trends.

Altogether we can conclude from this scenario that introducing just a single component can lead to very significant improvements for the entire system. Choosing the right factor makes it in both cases possible to obtain a global outcome that is halfway in between pure ROP and complete CPFRI system performance.

Two components evaluation
At this stage we combine two out of the four CPFRI components. Thus six possible combinations of components can be derived and will be evaluated for their impact on the entire replenishment system.

The most apparent finding when looking at the two-component table results for Company 1 is the diversity of achieved improvements.

Table 4: Supply Chain performance figures for two-component evaluation Company 1

<table>
<thead>
<tr>
<th>2-component Simulation Scenario Company 1</th>
<th>ROP</th>
<th>PA</th>
<th>SF</th>
<th>TF</th>
<th>ID</th>
<th>CPFRI</th>
</tr>
</thead>
<tbody>
<tr>
<td>Overall SL Gap</td>
<td>5.78</td>
<td>1.60</td>
<td>3.18</td>
<td>5.45</td>
<td>1.40</td>
<td>1.51</td>
</tr>
<tr>
<td>SL gap Customer 1</td>
<td>4.0</td>
<td>0.7</td>
<td>0.4</td>
<td>0.6</td>
<td>1.0</td>
<td>0.0</td>
</tr>
<tr>
<td>SL gap Customer 2</td>
<td>0.1</td>
<td>0.6</td>
<td>0.2</td>
<td>0.6</td>
<td>1.0</td>
<td>0.0</td>
</tr>
<tr>
<td>SL gap Customer 3</td>
<td>0.6</td>
<td>0.2</td>
<td>0.2</td>
<td>0.6</td>
<td>1.0</td>
<td>0.0</td>
</tr>
<tr>
<td>SL gap Customer 4</td>
<td>1.2</td>
<td>0.8</td>
<td>0.8</td>
<td>1.2</td>
<td>1.6</td>
<td>2.0</td>
</tr>
<tr>
<td>Median SL gap</td>
<td>18.4</td>
<td>8.2</td>
<td>14.7</td>
<td>15.3</td>
<td>7.1</td>
<td>6.5</td>
</tr>
<tr>
<td>Largest gap</td>
<td>100.0</td>
<td>77.1</td>
<td>68.1</td>
<td>91.5</td>
<td>47.7</td>
<td>45.1</td>
</tr>
<tr>
<td>Weeks of p.s.</td>
<td>77.9</td>
<td>85.5</td>
<td>81.0</td>
<td>72.6</td>
<td>82.0</td>
<td>75.1</td>
</tr>
<tr>
<td>ProdfC Accuracy</td>
<td>97.4</td>
<td>96.1</td>
<td>93.8</td>
<td>97.0</td>
<td>95.0</td>
<td>94.5</td>
</tr>
<tr>
<td>Uncritical delays</td>
<td>43.4</td>
<td>32.4</td>
<td>40.6</td>
<td>51.9</td>
<td>30.9</td>
<td>41.1</td>
</tr>
<tr>
<td>Critical delays</td>
<td>15.8</td>
<td>8.6</td>
<td>15.5</td>
<td>36.5</td>
<td>8.0</td>
<td>17.1</td>
</tr>
<tr>
<td>Fill rate Manuf</td>
<td>85.6</td>
<td>90.8</td>
<td>90.8</td>
<td>84.4</td>
<td>90.1</td>
<td>87.9</td>
</tr>
<tr>
<td>Perfect delivery</td>
<td>43.3</td>
<td>52.5</td>
<td>46.7</td>
<td>41.4</td>
<td>54.4</td>
<td>49.5</td>
</tr>
<tr>
<td>Fill rate DCs</td>
<td>98.1</td>
<td>98.9</td>
<td>98.4</td>
<td>97.9</td>
<td>98.0</td>
<td>98.1</td>
</tr>
</tbody>
</table>

Looking at the overall service level gap as the most important measure shows results that constitute in the worst case almost no improvements from the ROP scenario (5.45%, Promotional data & intelligent delivery) but in the best case major progress towards full CPFRI achievements (1.4%. Seasonal factors & trend forecast). These results apply very similarly to the delay and other metrics as well. Particularly a combination of PAID is distinctly inefficient and thus not advisable.

Table 5: Supply Chain performance figures for two-component evaluation Company 2

<table>
<thead>
<tr>
<th>2-component Simulation Scenario Company 2</th>
<th>ROP</th>
<th>PA</th>
<th>SF</th>
<th>TF</th>
<th>ID</th>
<th>CPFRI</th>
</tr>
</thead>
<tbody>
<tr>
<td>Overall SL Gap</td>
<td>2.81</td>
<td>2.12</td>
<td>1.16</td>
<td>1.04</td>
<td>0.87</td>
<td>0.63</td>
</tr>
<tr>
<td>SL gap Customer 1</td>
<td>1.8</td>
<td>1.8</td>
<td>0.6</td>
<td>0.6</td>
<td>1.7</td>
<td>0.4</td>
</tr>
<tr>
<td>SL gap Customer 2</td>
<td>4.7</td>
<td>3.2</td>
<td>1.9</td>
<td>1.8</td>
<td>1.6</td>
<td>0.9</td>
</tr>
<tr>
<td>SL gap Customer 3</td>
<td>4.7</td>
<td>3.2</td>
<td>1.9</td>
<td>1.8</td>
<td>1.6</td>
<td>0.9</td>
</tr>
<tr>
<td>Fill rate Manuf</td>
<td>95.8</td>
<td>97.6</td>
<td>86.0</td>
<td>93.5</td>
<td>95.8</td>
<td>93.7</td>
</tr>
<tr>
<td>Fill rate DCs</td>
<td>90.5</td>
<td>96.8</td>
<td>98.9</td>
<td>98.7</td>
<td>98.9</td>
<td>99.1</td>
</tr>
</tbody>
</table>

The level of diversity between the several component combinations is just as high for Company 2 as it was for Company 1. As it was already the case for the one-component evaluation the particular importance of the various factors differs between the two scenarios. In case of Company 2 Trend-Forecast and Promotional activity data is of major importance whilst especially seasonal factors seem to be of limited use. As a nevertheless slight surprise the “winning” setting for this case is the combination of trend forecast and “intelligent” delivery that achieves a remarkably good outcome coming very close to fully-featured CPFRI regarding service level and delay figures. Altogether four out of six settings are very much recommendable whilst only two disappoint. Comparing the results from the best possible cases from both companies’ scenarios with the reference full-featured CPFRI figures makes it obvious that it is possible to achieve nearly CPFRI quality outcome with only two out of four components enabled. Thus a company being aware of the crucial aspects within their demand behaviour and thus supply chain delivery structure can save considerable amount of money and implementation effort in focusing only on the
components that are most useful for them. On the other hand the two tables above show that taking the wrong components can result in an even inferior outcome compared to the pre-implementation phase. If for example Company 1 would spend vast effort in obtaining accurate promotional schedules and demand impact data together with establishing an intelligent delivery system that handles deliveries on the basis of urgency-levels – service level gaps would diminish only slightly whilst fill-rates stay the same and delivery delay figures get even worse than before. This would render the implementation effort and cost virtually worthless. The same would happen if Company 2 decided for a detailed seasonal factor analysis and an intelligent delivery system to improve their delivery/replenishment process. The overall achievements in these cases would most likely be very disappointing.

Three component evaluation
As a final step we will also have a look at the 3 component scenarios. There are four possible combinations that will be compared with the 4-component, full-featured CPFR setting.

Table 6: Supply Chain performance outcome within three component evaluation

<table>
<thead>
<tr>
<th>Simulation Scenario Company 1</th>
<th>PA6-STFT</th>
<th>PA6-STFD</th>
<th>PA7-STFT</th>
<th>PA7-STFD</th>
<th>CFFR</th>
<th>CPFR</th>
</tr>
</thead>
<tbody>
<tr>
<td>Overall SL Gap</td>
<td>5.6%</td>
<td>0.9%</td>
<td>1.1%</td>
<td>3.3%</td>
<td>4.4%</td>
<td>3.6%</td>
</tr>
<tr>
<td>GAP</td>
<td>2%</td>
<td>0.3%</td>
<td>0.5%</td>
<td>1.6%</td>
<td>2.1%</td>
<td>1.5%</td>
</tr>
<tr>
<td>Median SL Gap</td>
<td>0.1%</td>
<td>0.2%</td>
<td>0.3%</td>
<td>0.4%</td>
<td>0.5%</td>
<td>0.4%</td>
</tr>
<tr>
<td>Uncritical delays</td>
<td>4.0%</td>
<td>2.5%</td>
<td>3.7%</td>
<td>4.9%</td>
<td>5.0%</td>
<td>4.6%</td>
</tr>
<tr>
<td>Critical delays</td>
<td>0.9%</td>
<td>0.7%</td>
<td>1.2%</td>
<td>0.8%</td>
<td>0.9%</td>
<td>0.5%</td>
</tr>
<tr>
<td>Fill rate DC</td>
<td>98.1%</td>
<td>99.2%</td>
<td>99.1%</td>
<td>98.0%</td>
<td>98.3%</td>
<td>99.3%</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Simulation Scenario Company 2</th>
<th>PA6-STFT</th>
<th>PA6-STFD</th>
<th>PA7-STFT</th>
<th>PA7-STFD</th>
<th>CFFR</th>
<th>CPFR</th>
</tr>
</thead>
<tbody>
<tr>
<td>Overall SL Gap</td>
<td>2.8%</td>
<td>0.0%</td>
<td>0.2%</td>
<td>0.6%</td>
<td>0.4%</td>
<td>0.4%</td>
</tr>
<tr>
<td>GAP</td>
<td>1.2%</td>
<td>0.6%</td>
<td>1.6%</td>
<td>1.8%</td>
<td>1.9%</td>
<td>1.8%</td>
</tr>
<tr>
<td>Median SL Gap</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
</tr>
<tr>
<td>Uncritical delays</td>
<td>4.4%</td>
<td>6.7%</td>
<td>3.7%</td>
<td>4.1%</td>
<td>3.6%</td>
<td>3.6%</td>
</tr>
<tr>
<td>Critical delays</td>
<td>0.7%</td>
<td>0.4%</td>
<td>1.2%</td>
<td>0.8%</td>
<td>0.9%</td>
<td>0.9%</td>
</tr>
<tr>
<td>Fill rate DC</td>
<td>97.1%</td>
<td>92.2%</td>
<td>94.0%</td>
<td>92.1%</td>
<td>93.2%</td>
<td>92.4%</td>
</tr>
</tbody>
</table>

At this stage basically all performance measures show significant improvements compared to the initial ROP scenario which does not come as a surprise. Although each setting features three out of four CPFR components the resulting outcome is still very heterogeneous. The main outcome of this scenario should be to basically find out what particular CPFR components are most negligible considering implementation. In case of Company 1 we can see that almost full CPFR like results are achievable with ignoring either the intelligent delivery system or detailed promotion activity information. Company 2 on the other hand could ignore either detailed seasonal factor analyses or the intelligent delivery system.

Hence we can see that a company thinking about putting a CPFR replenishment system into working can get valuable information about what components are essential for a successful implementation and which ones can rather be neglected. On the other hand we can see which component is the actual most important even in combination with any of the other three. In case of Company 1 seasonal factor analysis is absolutely vital whilst this holds true for the short/mid-term trend forecast for Company 2. Any three component combination including these factors leads to very satisfactory results whilst only the ones not including them disappoint. Nevertheless the analysis also shows that all four components are certainly necessary to obtain the absolute best results. Each individual result of the full-featured CPFR system is superior to every 3-component outcome. Thus we may argue that certain components are more or less important than others but in the end they all contribute to some extent to an overall superior outcome.

CONCLUSIONS

The previous analysis revealed interesting insights into the distinct performance contributions of four chosen typical CPFR components. These were evaluated using simulation of supply chain frameworks of two distinct mid-size food manufacturers. Whilst the particular recommendations about which component is somehow superior over another varied due to the heterogeneity of the individual demand behaviour, interesting conclusions were found about the possibility to choose certain components prior to any CPFR process implementation. The investigation clearly showed that remarkable efficiency advances can be achieved to a very superior level with basically only two out of four available components as long as only the right ones were chosen. This should encourage companies to evaluate their delivery/demand systems as to what extent it is necessary to implement and maintain all stages of a considered CPFR system. Depending on the individual kind and level of performance improvement that is wanted, a given company should choose according to the investigation results, which of the good scenarios is most easy and cost-effective to implement. Thus a detailed prior investigation is very much recommended to save cost and effort and avoid disappointing collaboration experiences.

The positive impact of this kind of framework analysis could be experienced within the cooperation with the two companies involved in this analysis. Both welcomed the obtained information which proved to be very valuable within the definition and the negotiation process of their future collaboration outline.

Limitations of the investigation

Besides the various interesting insights that could be obtained by individually combining several CPFR components it has to be mentioned that the analysis itself very much focuses on the circumstances of the two involved manufacturers. Companies with dissimilar demand and delivery structures should not judge about the general importance of any particular component based on only this study. Depending on the intensity of promotional activities regarding frequency and impact, the level of
seasonal sales/demand deviations, the order-behaviour/discipline of customers or the extent of short-term demand fluctuations, every one of the above mentioned four components could theoretically be the least or most important one in any other, somewhat different, supply chain environment. For the case of the traditional ROP ordering procedure rational ordering is assumed which excludes adaptive ordering behaviour according to actual received quantities (e.g. bullwhip effects). Thus a lower load-level/fill-rate will not result in artificially high adjusted demand (no “shortage-gaming”) as it might be the case in reality. Another limitation constitutes the choice of CPFR components. A CPFR system usually includes somewhat more complexity than incorporated by the four taken factors. Aspects like communication advantages due to joint business forecast and thus additional informal connections could not be modelled as well as an extended level of variation that is typical for a supply chain that constantly has to adapt to new external influences. The investigated product-groups have furthermore a considerably stable demand behaviour which makes modelling more reasonable but limits the transferability of results to different kinds of products or even industries. Looking at the above findings and considering the limitations of the investigation it would be interesting for further research to carry out our analysis while varying the above assumptions and conduct more case studies to approve or adjust the previous conclusions as to give practitioners further ideas on how to benefit from evaluating individual CPFR components for their particular use in certain supply chain environments.

REFERENCES


LINE AVAILABILITY AND MATERIAL SYNCHRONIZATION IN LIQUID PACKING LINES: A SIMULATION STUDY

Steven Moerman(1), Jaap Ottjes(2), Mitch Gorsira(1), Gabriel Lodewijks(2)

(1) Procter & Gamble
European Technical Centre
Temsealaan 100, StrombEEK-BeVER,
B-1853 Belgium
Gorsira.m@pg.com

(2) Delft University of Technology
Faculty of Mechanical Engineering and Marine Technology
Mekelweg 2, 2628 CD Delft, The Netherlands
J.A.Ottjes@tudelft.nl

KEYWORDS
Line availability, changeover losses, material losses, produce to demand, packing lines

ABSTRACT
Increasing the number of product variants in line production usually shortens the average batch run lengths and consequently increases the number of changeovers. If new equipment has to be installed the line performance may change. In this paper a case is discussed concerning the modification of a production line of bottles with liquid due to the introduction of a new product variant. Simulation has been used to determine the new line availability, to assess the changeover losses and to improve material synchronization.

INTRODUCTION
A significant trend in the Fast Moving Consumer Goods business is the increase in the number of product variants, accompanied by a decrease in production volume per variant. The result is an increased complexity in the production environment. In case of the introduction of a new product variant, additional machines might be necessary on the production line. An increasing number of product variants will trigger a movement from “produce to stock” to “produce to demand”.

In a “produce to stock” environment, product variants are made in large production runs, based on a demand forecast. Changeover of the production line occurs less frequently and is therefore less dominant in production parameters. In a “produce to demand” environment, short production runs are executed of every variant, mainly to keep inventory low. Frequent changeover of the production lines is necessary then, but this may contribute significantly to production losses, both time losses and material losses.

For liquid packing lines, it is difficult to control the machines in such a way that the production run is exactly ended at the desired amount of produced items. This may lead either to partly finished pallets causing excess storage and handling cost or to scrap cost. These effects will increase in the case of “produce to demand”.

For a new liquid product variant in a “produce to demand” environment, which requires additional machines on the production line, the following research questions need to be addressed:

- What will be the impact of the new machines on the existing production line?
- How can the changeover losses be minimized?
- How can the material flows be controlled in the end of a production run?

This paper will discuss the results of a project concerning a production line of cleaning liquid, that has to be modified to enable the production of a new variant. Simulation is used to answer the questions above. First, the case will be introduced. Then the simulation environment will be discussed, followed by the experiments and results and finally the conclusions will be drawn.

CASE DESCRIPTION
The products concerned are bottles filled with a cleaning liquid. The bottles are filled, labeled and packed at the packing floor, from logistical point of view the most interesting part of the plant.

![Figure 1: Schematic drawing of the plant](image-url)
The packing floor is fed by the liquids making department, the bottle blowing department and the packing material storage department. This is shown in Figure 1.

On the packing floor, several packing lines process material. The general lay-out of a production line is shown in Figure 2.

![Diagram of production line](image)

**Figure 2:** Schematic drawing of production line

ULF stands for Unit Load Former (palletizer). After the bottles are produced, the unscrambler aligns them on a conveyor. After the unscrambler, the bottles are labeled, filled and put in a case. The cases are palletized. The machines are connected by conveyors. The conveyors are short and therefore their buffer function is limited. Their primary goal is transport between the machines, not decoupling the machines. The filler machine (also capper) is regarded the bottleneck machine in this production line. Machine rates upstream and downstream of the filler increase stepwise.

Due to different causes machines will fail now and then, interrupting the flow of material. When a machine is down it will take some time to restart it and different failure modes will cause different downtimes. To describe the machine start/stop behavior, the following two expressions are used:

\[
\begin{align*}
\text{MTBF} & = \text{Mean Time Between Failures} \\
\text{MTTR} & = \text{Mean Time To Repair}
\end{align*}
\]

\[
\begin{align*}
\text{MTBF} & = \frac{\text{Total Uptime}}{\text{Number of Failures}} \\
\text{MTTR} & = \frac{\text{Total Downtime}}{\text{Number of Failures}}
\end{align*}
\]

To describe the performance of the production line, the concept of availability is used. For a period of undisturbed production, without planned downtime, the availability is defined as follows (Smith, 1981):

\[
\text{Availability} = \frac{\text{Uptime}}{\text{Total Time}}
\]

When the bottleneck machine of a production line has a constant rate, the availability for a period without planned downtime can also be defined as follows:

\[
\text{Availability} = \frac{\text{Actual Production}}{\text{Potential Production}}
\]

If different product variants are to be produced on the same line, and so the line has to be modified between variants: a changeover. The changeover losses can be divided in two categories. The first concerns a loss of time: during a changeover there is no production possible. The second loss concerns material. At the end of a production run there is always material (bottles, labels, caps, liquid, cases, pallets) on the line that has to be scrapped or stored. The storage of partly filled pallets increases costs.

In practice the system is not able to track every individual bottle on the line and therefore it is not possible to end the production run exactly at the desired production volume. If there is only a small amount of material left at the end of a run, it will be scrapped. If there is a lot of material left, it will be stored and reused at a later run of the same “stock keeping unit” (SKU). Given the trend towards ‘Produce to Demand’ and the accompanying increase in changeovers, there is a great need for improvement of this end of run procedure. One of the options is to (partly) automate the end of run procedure.

To answer the research questions, the following information is needed:
- The new line availability compared to the original
- Line availability as a function of filler MTBF
- Line availability as a function of filler MTTR
- Changeover losses as a function of number of operators
- Changeover losses as a function of average production run volume

**MODELING**

In the literature much work has been done on optimizing production control in the (semi) process industry (Günther and van Beek, 2003). When it comes down to adapting or extending a specific production environment, simulation appears to be a very powerful and flexible tool. Production in the (semi) process industry usually involves both continuous and discrete processes. Consequently simulation tools should be able to support both ways of production (Clark and Joglekar, 1992). Siererberg and Wever (Siererberg and Wever, 1982) simulated complete beer-bottling lines in a combined continuous-discrete model using the simulation language Prosim, one of the first advanced combined discrete-continuous simulation language (Prosim Web site 2006). This model was built to improve line productivity and used machine down time distributions derived form the real machines in the line. The simulation package used in this project is Extend (Extend web site 2006), a widely used tool within the company. The software package contains a wide variety of predefined blocks in libraries. The user can build its own model by selecting and connecting the appropriate blocks and providing the right control settings for the blocks. The program is easily linked to MS Excel for input and output files.
In the constructed model, a flow-based module is combined with an item-based module. The flow module represents the production line as displayed in Figure 2, and it communicates with the item-based changeover module. The changeover module is activated by signals from the production line, and then the changeover module indicates when the production line is ready to resume its activities.

The input for the model consists of:
- Machine Rates
- Machine Reliability Data
- Conveyor Lengths
- SKU Production Schedule
- Number of Operators
- Changeover Times per Machine

The reliability data of the machines consists of MTBF’s and MTTR’s. For the existing machines, real reliability data can easily obtained from the company’s records. For a new machine, benchmarks with comparable machines combined with an expert opinion provide a good first indication. A sensitivity analysis for deviation of this indicated value was executed.

The production schedule includes the sequence and quantities of the different SKU’s. Depending on the from/to SKU, the model detects which tasks have to be executed for each changeover and acts accordingly. The simulation runs are executed with a production schedule of one month. The number of operators available can be varied per changeover or can be varied over time to resemble lunch break and shift change.

The Production line module

In the production line module the machines are the active elements, and the bottles and the liquid are passive elements, processed by the machines. The conveyors in between machines and the hopper are modeled as ordinary FIFO queues. Every machine processes material, but is now and then interrupted because of a machine stop. Every machine has a downtime distribution and an uptime distribution, based on the given MTTR and MTBF.

Process bottle blower, case-erector and CLP:

IF downstream queue is NOT full,
THEN process material at prescribed rate
ELSE hold

All machine have fixed rates, except for the CLP. This rate is depending on the surge tank level:

CLP rate:
- IF surge tank level is < target
  THEN CLP speed is highCLPspeed
- ELSE CLP speed is lowCLPspeed

For the unscrambler, the labeller, the filler, the casepacker and the ULF, the processes are identical, however the CLP can only work when both upstream queues are not empty (liquid and bottles). And for the ULF the downstream queue will never be full, as the warehouse is assumed infinity large.

Process unscrambler, labeller, filler, packer and ULF:
- IF downstream queue is NOT full,
  AND upstream queue is NOT empty
  THEN process material at prescribed rate
- ELSE hold

This describes the basic behavior of the production line module. The number of bottles with liquid ‘flowing’ through the line depends on the machine rates and the start-stop behavior of the machines.

The Changeover Module

In this module, the operators are the active elements, and the machines are the passive elements. This part of the model is item based. Based on a signal from a machine, indicating that the target production for this SKU is achieved, the machine will stop and will be marked ‘ready for changeover’. Next an operator will prepare the machine for the time needed for a changeover (can be zero), and the machine will be marked ‘ready for production’. The (variable) pool of operators will work their way downstream from the bottle blower to the ULF. When all machines are marked ‘ready for production’, the production module will commence producing the next SKU. The influence of the number of operators on the changeover time is illustrated in Figure 4.

![Figure 4: Operators influence on changeover time](image)

<table>
<thead>
<tr>
<th>OPERATOR 1:</th>
<th>TASK 1</th>
<th>2</th>
<th>3</th>
<th>TASK 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>TOTAL = 26 mins</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>OPERATOR 1:</th>
<th>TASK 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>OPERATOR 2:</td>
<td>2</td>
</tr>
<tr>
<td>TOTAL = 16 mins</td>
<td></td>
</tr>
</tbody>
</table>
The influence of the number of operators on the changeover is depending heavily on the tasks that have to be executed in a changeover, and thus on the 'From' and 'To' SKU type of changeover. One of the tasks of the changeover model is that it detects the 'from' and 'to' SKU and then determines the tasks to be executed.

EXPERIMENTS AND RESULTS

For the new product variant, one of the additions to the production line is a new filler/capper machine. The system availability is lower with this machine than without it, since the machine process is quite complex. The influence of unexpected behavior of the machine on the system availability is determined. In Figures 5a and 5b, the results are displayed for filler behavior deviating from the expected. The numbers are indexed for confidentiality reasons.

![Figure 5a: System Availability vs. Filler MTTR](image)

System Availability vs. Filler MTTR

![Figure 5b: System Availability vs. Filler MTBF](image)

System Availability vs. Filler MTBF

The open dots represent the expected situation, the closed dots represent what-if scenarios. As could be expected, influence of the filler behavior on system availability is quite significant, as it is the bottleneck machine. From the graphs we read that, from operational point of view, it is most efficient to first make sure the filler MTBF is high enough, before focusing on ways to decrease the filler MTTR.

The next question to answer regards the changeover time losses. Since in a "produce to demand" environment, the changeovers will occur more frequently, it is important to develop ways of reducing changeover losses. One option is increasing the number of operators that are executing the changeover. Figure 6 displays the relation between changeover losses and the number of operators.

![Figure 6: Changeover Losses vs. Number of Operators](image)

Figure 6: Changeover Losses vs. Number of Operators

The research clearly shows that adding extra operators does reduce changeover losses, until a certain limit. This limit is governed by the longest individual task in a changeover. In this specific case, 4 operators would already deliver the minimal changeover loss.

The other aspect of “produce to demand”, shorter run lengths, also influences the changeover losses, since in a certain period of production, more time will be consumed by changeover tasks.

![Figure 7: Changeover Losses vs. Average Run Length](image)

Figure 7: Changeover Losses vs. Average Run Length

Figure 7 shows that the changeover time losses will increase faster than linear when the average run length decreases.

The next step is to find a way to minimize the material losses at a change over. Essential for ending the production run without material on the line and exactly the desired produced volume, is shutting down the machines at the correct point in time. Before the project started, shutting down machines was done by hand by the operator. There was a large risk of human errors, for example if operators are distracted on critical moments. As part of the project,
the packing line PLC was extended to include automated shutdown of the unscrambler and the case-erector.

The decision to shut down a machine should be based on how many bottles (on spec) have passed that machine. The difficulty is that if x bottles have passed a machine, not necessarily x bottles will make it to the end of the line; at various points in the line scrap will occur at unpredictable rates. Therefore another approach is chosen. Starting point is a counter of the number of bottles that have made it to the end of the line. Whenever this counter, combined with the number of bottles on the line, adds up to the desired production volume, then the first upstream machine is shut down. The absolute margin of error (because of scrap) is much smaller this way because the margin is not derived from the total produced volume, but only from the number of bottles on the line.

The difficulty now is to assess the number of bottles on the line, since this is not a constant and it is also not measurable. With simulation it is possible to determine the average number of bottles on the line during constant production. The graph in figure 8 is constructed using simulation results.

The number of bottles on the line will vary between a minimum (all conveyors empty) and a maximum (all conveyors full). The variation is caused by the machine characteristics: when a machine fails, the upstream conveyors will fill up and the downstream conveyors will run empty. So the distribution of Figure 8 is governed by the machine reliability characteristics. The highest peak corresponds with normal operation, all other peaks are irregularities. The distribution is different for all lines, and for all bottle sizes.

If a certain number of bottles on the line is assumed (μ) there is a risk of undershooting this value. This means that the run will end with a partly finished pallet which has to be handled and stored. On the other hand, in case of overshoot of the estimate, there is a risk of scrap. If the distribution of bottles on the line is represented by a normal distribution, the balance between storage and scrap would look like Figure 9.

![Figure 8: Probability of number of bottles on the line](image)

The bumpy character of the storage cost line is caused by the distribution of bottles on the line. With this cost evaluation it is possible to set the estimate of number of bottles on the line to the point where the total of storage cost and scrap cost is minimized.

CONCLUSIONS

Simulation enabled an up front analysis of the effect of adding extra machines to an existing production line. Several what-if scenarios could be analyzed. The results strengthen management’s earlier findings to focus first on increasing a machine’s MTBF, before increasing the system productivity further by decreasing the MTTR.

In the simulation model, a flow-based module was successfully combined with an item-based module. They interacted via signals to simulate the production line (flow-based) and the changeover procedure (item-based).

With regard to the changeover losses, the following is noted. It makes sense that a more “produce to demand”
oriented production environment incurs higher changeover losses. The experiments show that measures can be taken to reduce the changeover losses. However, it makes no sense to start reducing these losses without a holistic view. The costs of these measures should always be balanced against the advantages in other parts of the supply chain. Produce to demand also delivers significant advantages, which might easily overshadow an increase in changeover losses.

For the material synchronization, it is concluded that simulation is a powerful tool to statistically analyze a complex problem. Simulation provides the necessary input data for a significant improvement in material synchronization. The simulation results indicate that, at the same scrap level, a 70% decrease in partly finished pallets is feasible. The findings of this study are used in the design of new production lines.

REFERENCES


Extend web site 2006: http://www.imaginethatine.com


Prosim Web site 2006: http://www.prosimbv.nl


EFFECTIVE JOB SHOP PRIORITY SCHEDULING
Hans P.M. Veeke, Gabriël Lodewijks
Delft University of Technology, Faculty 3mE
Mekelweg 2, 2628 CD Delft
H.P.M.Veeke@tudelft.nl

KEYWORDS
MRP planning, job scheduling,
Job shop simulation

ABSTRACT
Job shop planning and scheduling have been thoroughly investigated during the last decades. The main research question was to determine the optimal load and sequence of jobs with limited resources available. Job shop environments are very stochastic (resources break down; customer orders are added at the last moment etc.). Because of these stochastics, the planning is usually based on statistical data from the past. On the contrary, scheduling has to sequence individual jobs, for which statistics can not be used. There is a natural friction between statistical decision making and individual job scheduling.

In this paper the effects of some basic scheduling priority rules are illustrated which lead to the conclusion that freedom of action (i.e. there is no detailed production sequence of tasks) is a promising alternative for effective job scheduling during manufacturing. It is shown that planning decisions can be based on sets of jobs, for which the statistics hold. This combination of planning and scheduling already results in controlled throughput times and gives a clear basis for decision support during manufacturing.

INTRODUCTION
Usually Enterprise Resource Planning (ERP) systems use basic priority rules for the delivery control in job shop environments (see Russell, Taylor, 2006). Applying these rules is the last step of an MRP-based approach, starting with global and detailed load balancing. The objective of job shop scheduling is assumed to be finding a trade off between loading efficiency and delivery accuracy (Kemppainen, 2005). The resulting production schedule is a plan for what part will be made at what time utilizing what resource (Parsons, Phelps, 2001). Planning and scheduling have become even more important due to the increased pressure on throughput times. Nowadays it is necessary to deliver in short time and to react in a flexible way to changing market demands.

In this paper an alternative way is presented to schedule customer orders in a job shop production. The primary goal of the research is to support decision making during the planning phase. Rather than shortening throughput times, the starting point for research was to achieve throughput time control.

Controlling throughput time is considered the first requirement in order to achieve delivery reliability. If the required level of reliability is reached, then the next step will be to shorten throughput times.

First, a number of aspects, which influence the results of priority rules for scheduling, will be investigated. The first experiments can be considered an extended verification of the simulation model. Based on the results, an alternative approach will be proposed, which shows promising results for achieving delivery reliability with real support for decision making, and sufficient flexibility to react on disturbances. This approach preserves high occupancy values.

The simulation model used, models an existing production company with a characteristic order flow.

THE SIMULATION MODEL
The simulation model describes a company that makes highly specialized optical instruments on customer order. Every order triggers 'Material Supply' to provide the required materials from stock or suppliers. 'Parts Manufacturing' then produces the required parts and 'Assembly' finally assembles these into the ordered products. 'Parts Manufacturing' is functionally organized (job shop). There are three departments: ‘Milling’, 'Fitting' and a ‘NC-department’. Every part to be manufactured should be processed (eventually several times) by several departments. The parts are being processed in batches. The batch size depends on the customer order size. On average a product consists of 2 components.

The model has been programmed in TOMAS (Veeke, Ottjes, 2000), an object oriented simulation extension of Delphi.

PRIORITY RULES
The simulation model contains three default scheduling alternatives:

- First Come First Served (FCFS): selection sequence depends on the arrival sequence
- Shortest Processing Time first (SPT): the batch with the smallest processing time is selected. It is well known that SPT shortens the average throughput time, but increases the standard deviation. To compensate for this, the batches are divided into two classes: batches with a task time smaller or larger than some value P. Batches with a task time smaller than P have
priority to the other batches, but are mutually scheduled according the FCFS-rule. The larger batches are scheduled in sequence of increasing task time. By varying \( P \) it is possible to apply the SPT-rule to some extent. If \( P = 0 \), the usual SPT-rule is applied, and if \( P = \infty \) the FCFS-rule applies.

- Operation Due Date (ODD): the batch with the shortest starting time is selected.

Although these rules have been investigated intensively, the effects depend on the task time distributions, which in this research are directly derived from reality. The results of the simulation runs are shown in table 1. These results include delays in material supply.

The high standard deviation causes a low planning reliability. The table shows clearly that the average throughput time decreases significantly by using the SPT-rule. In the table SPT 98% means that 98% of the batches has a task time below \( P \). Still, the standard deviation remains large for each SPT-alternative and the batches with a large task time show a very slow progress (see column Max). However, the table also shows that the SPT-rule, where only 5% of the batches is selected according the SPT-rule (SPT 95%) already realizes 80% of the maximum gain in throughput time, which could be realized if all batches are selected according the SPT-rule.

<table>
<thead>
<tr>
<th></th>
<th>T</th>
<th>MDD</th>
<th>SDD</th>
<th>Min</th>
<th>Max</th>
</tr>
</thead>
<tbody>
<tr>
<td>FCFS</td>
<td>171</td>
<td>12.5</td>
<td>9.7</td>
<td>-3.1</td>
<td>39.5</td>
</tr>
<tr>
<td>SPT 98%</td>
<td>116</td>
<td>11.1</td>
<td>10.0</td>
<td>-3.0</td>
<td>73.0</td>
</tr>
<tr>
<td>SPT 95%</td>
<td>109</td>
<td>11.0</td>
<td>10.0</td>
<td>-3.0</td>
<td>73.0</td>
</tr>
<tr>
<td>SPT 0%</td>
<td>91</td>
<td>10.5</td>
<td>10.0</td>
<td>3.1</td>
<td>72.5</td>
</tr>
<tr>
<td>ODD</td>
<td>173</td>
<td>11.5</td>
<td>7.9</td>
<td>2.9</td>
<td>27.2</td>
</tr>
</tbody>
</table>

Table 1. Effect of FCFS-, SPT- and ODD-rules.

\( T \) = throughput time (hours)

MDD/SDD: Mean/Standard Deviation of Delivery Week (weeks)

**SUPPLY DELAYS**

Two factors determine the actual delivery time:

a. the performance of material supply
b. the production process itself

Both factors are a source for a planning delay. The parts belonging to one single component to be assembled, should be delivered at the same time. So a component consists of a set of parts, each requiring different materials that are delivered by different suppliers. The last supplier determines the delay with respect to final delivery. In the model, on average 2 suppliers are involved for the parts manufacturing for one component. The supply patterns for all suppliers both at the part and the component level are shown in figure 1.

![Supply reliability](image)

Figure 1. Delivery pattern Material Supply

On average the materials for each separate part are delivered 4 weeks late. Measuring the material delivery for each component, then the delivery is on average 9 weeks late. So the major part of the delay in component delivery is caused by part delivery. From now on this delay is removed from the model and the assumption is made that all materials are delivered in time.

The delivery to Assembly shows a similar effect. The realisation of the planned start time of Assembly depends on the last arrival of parts from Parts Manufacturing. If all parts separately are being delivered in time on average, then the set of parts for one component will be late on average. There are two ways to prevent this effect:

- each part should be planned “early”
- the standard deviation around the average should be reduced to (almost) zero.

The first solution would increase the stock of parts significantly and the second solution will be impossible for a stochastic job shop organisation. This research will focus on reduction of the deviation and investigate for this the influence of the order characteristics and the way of planning itself.

**TASK TIMES**

The variation in task times is large (see table 3). In reality they vary between 1 and more than 80 hours. It makes sense to investigate the effect of long task times, because these times particularly cause long waiting times (for this reason the SPT-rule is often used).

<table>
<thead>
<tr>
<th></th>
<th>0-4</th>
<th>4-8</th>
<th>8-16</th>
<th>16-40</th>
<th>&gt;40</th>
<th>M</th>
</tr>
</thead>
<tbody>
<tr>
<td>Milling</td>
<td>22%</td>
<td>28%</td>
<td>27%</td>
<td>17%</td>
<td>6%</td>
<td>13.5</td>
</tr>
<tr>
<td>NC</td>
<td>7%</td>
<td>22%</td>
<td>33%</td>
<td>12%</td>
<td>26%</td>
<td>25.0</td>
</tr>
<tr>
<td>Fitting</td>
<td>51%</td>
<td>21%</td>
<td>14%</td>
<td>11%</td>
<td>3%</td>
<td>8.7</td>
</tr>
</tbody>
</table>

Table 3. Task times (hrs) per department

\( M = \) Mean task time

In the model, the longest task time is assumed to be 80 hours. In order to investigate the effect of extremely long task times, each task time larger than 40 hours is split into two tasks, one with a task time of 40 hours and one with
the rest of the task times. The planning takes this into account by scheduling both tasks sequentially with intermediate waiting times of 40 hours (see table 4). Splitting the task times has no significant effect. Although the average task time decreases, the average number of tasks increases.

<table>
<thead>
<tr>
<th></th>
<th>T</th>
<th>MDD</th>
<th>SDD</th>
<th>Min</th>
<th>Max</th>
</tr>
</thead>
<tbody>
<tr>
<td>Full</td>
<td>176</td>
<td>3.5</td>
<td>3.9</td>
<td>-2.1</td>
<td>20.9</td>
</tr>
<tr>
<td>Tasks</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Split</td>
<td>167</td>
<td>3.0</td>
<td>3.7</td>
<td>-2.5</td>
<td>23.4</td>
</tr>
</tbody>
</table>

Table 4. Effect of task times

**PLANNING**

Job shops try to achieve delivery reliability while maintaining a maximum occupancy (preferably 100%) of the available capacity. These goals are in essence conflicting. So usually one tries to achieve some optimal situation: acceptable reliability with acceptable occupancy. The occupancy of the 3 groups with normal task times is 80 to 90%, which completely consists of tasks that have been planned in detail. A number of experiments has been done to investigate the effect of adding tasks, which don’t have a planned delivery time, but are merely used to occupy capacity up to 100%. The results are shown in the table below.

<table>
<thead>
<tr>
<th></th>
<th>T</th>
<th>MDD</th>
<th>SDD</th>
<th>Min</th>
<th>Max</th>
</tr>
</thead>
<tbody>
<tr>
<td>P/U:  90 / 0%</td>
<td>176</td>
<td>3.5</td>
<td>3.9</td>
<td>-2.1</td>
<td>20.9</td>
</tr>
<tr>
<td>P/U: 70 / 30%</td>
<td>117</td>
<td>1.8</td>
<td>2.1</td>
<td>-2.2</td>
<td>14.5</td>
</tr>
<tr>
<td>P/U: 50 / 50%</td>
<td>106</td>
<td>1.5</td>
<td>2.2</td>
<td>-2.0</td>
<td>13.2</td>
</tr>
<tr>
<td>P/U: 50 / 50%</td>
<td>85</td>
<td>0.9</td>
<td>1.4</td>
<td>-9.2</td>
<td>9.2</td>
</tr>
</tbody>
</table>

Table 5. Effect of planned tasks vs. unplanned tasks P/U= %planned occupancy/ % unplanned Occ.

All tasks have a task time according the original task time distribution. In these experiments however, the unplanned work is only disturbing the progress of planned work. Although planned work receives priority over unplanned work, planned work may have to wait at a capacity group to complete an unplanned task.

The effect of this disturbance can be minimized by creating only short tasks for the unplanned work, as can be seen in the last row in table 5. In this experiment the tasks of unplanned work have a task time between 10 and 20 hours. The conclusion of these experiments is that decreasing the occupancy for planned work has a positive effect on delivery reliability. However, the complete workload of Parts Manufacturing usually consists of planned work. The major question is then: how to split the workload in planned and unplanned work.

**URGENCY AND PRIORITY**

Until now, the starting point has been a planning, generated according MRP principles. All experiments tried to realize this planning as good as possible. Usually, MRP plans the manufacturing tasks by establishing the starting times for each task. Above that, waiting times for each task are inserted in front of the actual task times. One assumes that in this way Parts Manufacturing is able to realise the delivery time to Assembly. The original goal of this level of deal is:

- to organize the process in such a way that each task will be processed on time (or exactly according to planning)
- to provide the criteria to value the state of the manufacturing process at any moment in terms of ‘early,’ ‘on time’ or ‘late’.

The validity of these assumptions will now be investigated. As an example, figure 2 shows the planned and real progress of one batch of parts.

Suppose the batch should be delivered at time D to

Figure 2. Planned and real progress of a batch

Assembly. The processing of the batch consists of 4 tasks: T1, T2, T3 and T4. MRP plans these tasks as shown by the top line and calculates that the required materials should be available at time S. At time C (e.g. during the next planning cycle) one observes that task T1 has not been started yet. The batch is valued as “late” on the basis of which the batch receives priority over batches that are valued early or on time. Experiments with the simulation model however show that without assigning priority values, a progress line (as the bottom line in figure 2) is just as likely as an assumed late delivery. The batch may also be delivered early without priority. The conclusion is that considering the batch as “late” at time C is premature, because one doesn’t know the progress of the future trajectory yet. The final delivery time is in fact the cumulative result of temporary delays and leads.

There is another risk to planning in such detail. During execution, there is a tendency on the shop floor to realize not only the process times exactly, but also the planned waiting times. This causes a large sensitivity for disturbances, while the waiting times were meant to minimize sensitivity.

Finally, this way of planning fixes so many moments in time that a deviation of these moments is likely to appear.
In practice this happens indeed, and consequently many tasks are marked “priority tasks”. By all this, the credibility of planning is seriously damaged. In order to plan correctly it is necessary to keep the original goal of manufacturing planning in mind: “deliver the parts in time at Assembly”. Only the final delivery time should be realized by Parts Manufacturing. Planning is a control function and should only lead to interventions if this delivery time can definitely not be reached anymore. MRP apparently does not satisfy this requirement.

The planned progress of the batch of figure 2 can also be represented differently, as shown in figure 3.

![Figure 3. Planning of batch progress](image)

Each batch is during its progress alternating in one of the following states:

- in process at a capacity group
- waiting for a capacity group

Figure 3 shows horizontally the time-axis and vertically the use of waiting time (as planned). The manufacturing process should be completed within the drawn parallelogram. At time D the batch should be delivered to Assembly, while Material Supply should have all materials available at time S. If all tasks can be processed immediately then the batch can be delivered to Assembly at time ED (“Earliest Delivery time”). If time D is to be reached then the processing should ultimately start at time LS (“Latest Starting time”) and process the tasks immediately after (LS will shift to the right as soon as a task has been completed). W is the total planned waiting time and from figure 3 it is clear that

\[
W = LS - S = D - ED
\]

Symbolically expressed, Parts Manufacturing should reach the right hand side of the parallelogram before time D. The dotted line in the parallelogram (the original MRP-planning) represents only one possibility to achieve this, but MRP forces this dotted line to be the single standard for manufacturing. Suppose planning would only use S, ED, LS and L to value the progress of batches, is it possible then to effectively control the delivery times to Assembly?

Table 5 already showed the effect of the percentage of planned and unplanned work. From now on, planned work will be defined as all batches which still have slack for reaching the final planned delivery. The work without slack will be called “urgent work”. This classification is clearly illustrated by figure 3. Periodically (in practice usually once a week), the batches with and without slack are listed. Batches without slack are the batches that will not reach the moment LS during the next week, batches that do reach this point during the next week do not have any slack left and are marked “urgent batches”. These urgent batches will be planned in detail. The planning of these batches will be realistic, because all work in progress is now divided in two classes; planned and urgent batches. The urgent batches represent a far lower capacity occupancy and by assigning them priority over planned batches it should be feasible to realize the required delivery time. Two conditions should be taken into account:

- the percentage urgent batches (in working hours) should not be too large. A value around 60% occupancy by urgent batches should be preserved.
- The result of this classification is influenced by the number of tasks to be performed. The parts of the components for this company only require two tasks on average. This number is too low to control the ‘urgency’ effectively. In practice the number of tasks usually is around 8 on average. Therefore the number of tasks in the simulation model is changed to 8 on average (varying between 1 and 15).

During the simulation runs, the classification period is one week. The planned work is scheduled according the FCFS-rule or ODD-rule (but now based on the final delivery time D), the urgent work according increasing LS. The results are shown in table 6. This new approach is called the Parallelogram Method (PGM)

<table>
<thead>
<tr>
<th></th>
<th>T</th>
<th>MDD</th>
<th>SDD</th>
<th>Min</th>
<th>Max</th>
</tr>
</thead>
<tbody>
<tr>
<td>FCFS</td>
<td>675</td>
<td>9.2</td>
<td>11.3</td>
<td>-11.7</td>
<td>46.5</td>
</tr>
<tr>
<td>ODD</td>
<td>639</td>
<td>6.4</td>
<td>7.8</td>
<td>-11.6</td>
<td>27.5</td>
</tr>
<tr>
<td>PGM</td>
<td>617</td>
<td>6.1</td>
<td>8.2</td>
<td>-11.7</td>
<td>27.2</td>
</tr>
</tbody>
</table>

Table 6. MRP versus PGM

The conclusion is that PGM shows the smallest mean delivery delay, although on average still late. In the simulation model, PGM used the sum of all waiting times as originally defined for use with MRP. The advantage of PGM however, is the applicability for decision support. PGM offers an objective way to value the progress of batches. As mentioned before, planning should preserve that the occupancy by urgent work does not (structurally) exceed approximately 60%. If it does, two possibilities are evident for use: outsourcing or working overtime. During the PGM-run of table 6 it appeared that on average 150 hours of urgent work were waiting at each capacity group. In further experiments the model was extended with outsourcing. During classification, the hours of urgent work were calculated.
there was at least one capacity group with more than 250 (or 200 in a next experiment) hours of urgent work, it is decided that all new orders are outsourced during the next week. The results are shown below.

<table>
<thead>
<tr>
<th></th>
<th>MDD</th>
<th>SDD</th>
<th>Min</th>
<th>Max</th>
</tr>
</thead>
<tbody>
<tr>
<td>PGM</td>
<td>617</td>
<td>6.1</td>
<td>8.2</td>
<td>-11.7</td>
</tr>
<tr>
<td>≥250</td>
<td>386</td>
<td>0.6</td>
<td>4.0</td>
<td>-13.7</td>
</tr>
<tr>
<td>≥200</td>
<td>374</td>
<td>0</td>
<td>4.3</td>
<td>-14.7</td>
</tr>
</tbody>
</table>

Table 7. PGM with outsourcing

Apparently, outsourcing using this criterion is successful. The outsourced work represented 5 and 10% of the working hours respectively, so the occupancy of the capacity groups dropped only slightly. The standard deviation however remains high, so further research is required to minimize this.

The comparison between MRP with ODD and PGM with outsourcing are summarized in table 8.

<table>
<thead>
<tr>
<th></th>
<th>MRP</th>
<th>PGM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Occupancy</td>
<td>85-90%</td>
<td>80-85%</td>
</tr>
<tr>
<td>Delivery</td>
<td>6.5 weeks late</td>
<td>On time</td>
</tr>
<tr>
<td>Deviation</td>
<td>Large (8 weeks)</td>
<td>Large (4 weeks)</td>
</tr>
<tr>
<td>Throughput time</td>
<td>16 weeks</td>
<td>9.5 weeks</td>
</tr>
</tbody>
</table>

Table 8 MRP compared to PGM

CONCLUSIONS

A number of (well known) priority rules using MRP-based planning, are examined in order to improve production control. None of these rules appeared to be satisfactory. Excluding the effect of delays in Material Supply, it is still impossible for Parts Manufacturing to realise planned delivery times. Also a change in workload composition showed no improvements. Only a significant decrease of workload that was planned in detail, resulted in an average delivery on time. The way of planning was investigated and it is shown that MRP lacks the tools to effectively intervene with production progress. An alternative approach, the parallelogram method, was primarily developed to realize the delivery time to Assembly. With this approach, better results can be achieved, if the workload can be classified in urgent and planned work. Above that, the method has the following advantages:

- the method is simple to use. Only periodically a classification should be made of all work in progress.
  Two classes are defined: planned and urgent batches based on a simple unambiguous criterion.
- The method offers more freedom of action to Scheduling. Only one priority rule applies: urgent batches have priority.
- The method supports decision making effectively, e.g. on outsourcing, working overtime

Further research is required:

REFERENCES


SIMULATION OF A LIG(A) PRODUCTION LINE

Pascal Meyer\textsuperscript{a}, Markus Arendt\textsuperscript{b}, Joachim Schulz\textsuperscript{c}

\textsuperscript{a} Institut für Mikrostrukturtechnik, Universität Karlsruhe
\textsuperscript{b} ANKA Commercial Service, Forschungszentrum Karlsruhe
\textsuperscript{c} Institut für Mikrostrukturtechnik, Forschungszentrum Karlsruhe
D-76021 Karlsruhe
E-mail: pascal.meyer@int.fzk.de

ABSTRACT
The aim of this work is the modelling and simulation of a manufacturing line using the deep x-ray LIGA process, which produces high precise metal parts. The deep x-ray LIGA process, which combines X-ray lithography with electroplating and molding, is a technique used worldwide for the fabrication of high precision and high aspect ratio microstructures. For example, direct LIG(A) gears in a watch movement represent a large step in the development of mechanical horology. The LIGA-technique, particularly its extreme precision and sidewall quality, are combined in a synergic way with the properties of metals, namely in the case of watchgears of almost nearly pure but sufficiently hard gold. Over the past few years, we have addressed LIGA’s barriers to market penetration by consistently and continuously focussing on making the technology repeatable and reliable throughout all process steps including the fabrication of the x-ray masks. Though LIGA is a substrate based process sequence, the potential for batch fabrication of LIG(A)-parts has not been addressed to date. The next step will therefore consist of increasing the production throughput and decreasing the cost. A fully automated fabrication line (FELIG) for direct LIG(A)-parts in a clean-room environment will be completed at the ANKA synchrotron within the next 3 years (2005-2008). We describe a simulation model of the line to make strategic decisions related to the factory.

INTRODUCTION
The LIGA technique, a German acronym consisting of the letters LI (Röntgen Lithographic meaning x-ray lithography), G (Galvanik meaning electroplating) and A (Abformung meaning molding) developed at the Karlsruhe Research Center, offers the possibility of manufacturing microstructures with high aspect ratios, high accuracy from a variety of materials (metals, plastics and ceramics). The direct deep X-ray LIG(A) technique offers both sidewalls quality better than 50nm in electroplated parts and a lateral reproducibility significantly less than 1\textmu m. Parts made by this process offer the long term functional integrity and reliability required for a mechanical movement. Moreover, this process offers no limitations to the creativity for almost any design. The major difference of the deep x-ray LIGA process compared to other lithography processes lies in the irradiation step. The latter using a synchrotron can be performed at only few sites around the world. A fully automated fabrication line (FELIG) for direct LIG(A)-parts in a clean-room environment will be completed at the synchrotron ANKA within the next 3 years (2005-2008) (Meyer et al. 2004). Instead of using commercially available software like ARENA\textsuperscript{®} (Kelton et al. 2003, www.arenasimulation.com) or SIMPROCESS \textsuperscript{®} (www.simprocess.com), a custom made Borland C++ simulation model including all the processes (irradiation, development, electroplating, ...) has been developed to look at the potentials of the line, to view the influence of the most important parameters and to make strategic decisions related to the factory (Aybar et al. 2002).
<table>
<thead>
<tr>
<th>Thickness in μm</th>
<th>200</th>
<th>900</th>
<th>1400</th>
<th>2300</th>
</tr>
</thead>
<tbody>
<tr>
<td>Beam-line parameter</td>
<td>(1)</td>
<td>(2)</td>
<td>(2)</td>
<td>(2)</td>
</tr>
<tr>
<td>Exposure dose in mA.min/cm</td>
<td>500</td>
<td>1400</td>
<td>1700</td>
<td>3000</td>
</tr>
<tr>
<td>exposure time in h</td>
<td>0.30</td>
<td>0.65</td>
<td>0.77</td>
<td>1.42</td>
</tr>
<tr>
<td>hub:28mm</td>
<td>0.75</td>
<td>1.64</td>
<td>1.95</td>
<td>3.63</td>
</tr>
<tr>
<td>hub:90mm</td>
<td>6</td>
<td>32</td>
<td>38</td>
<td>60</td>
</tr>
<tr>
<td>development time in h</td>
<td>2</td>
<td>8</td>
<td>10</td>
<td>17</td>
</tr>
<tr>
<td>without megasonic</td>
<td>72</td>
<td>/</td>
<td>/</td>
<td>/</td>
</tr>
<tr>
<td>with megasonic</td>
<td>/</td>
<td>96</td>
<td>120</td>
<td>200</td>
</tr>
</tbody>
</table>

Table 1: Fabrication parameters (resist used is PMMA) - hub represents the scanning distance - (1) indicates the ANKA beam-line LITHO 2-(2) indicates the ANKA beam-line LITHO 3

**THE DEEP X-RAY LIGA PROCESS**

In the first step, the absorber structure of a x-ray mask is copied into resist layers by using synchrotron radiation. Usually, two types of resist material are used: poly(methylmethacrylate) (PMMA) and SU-8. In the case of the former, with exposure, the molecular weight decreases and it becomes soluble in a developer. In the latter case, there is a polymerisation and the unexposed part is soluble. The spaces generated by the removal of the irradiated plastic material can be filled with metal by electroplating processes. In this way, a negatively patterned secondary metal structure, such as nickel, copper and gold, or alloys, such as nickel-cobalt and nickel-iron, is generated. The different steps (see figure 2) and data of the process and are listed in the following points:

- Resist technology. It consists of applying a resist onto a substrate. The standard positive resist is PMMA. The parameter are the thickness of the resist and its dimension.

- Irradiation and development. LIGA technology needs a synchrotron beam-line to perform the resist exposition and a scanner to move the sample. The mask and the substrate holder are one of the parameter of the process. Substrates could be: 4-inch wafers (Si, SiO₂, Al₂O₃) or 6-inch wafers. In the copy step of the LIGA process, the deposited dose calculations in the resist are needed. The basic calculi needed for synchrotron beam-line design are related to the spectra characteristics and to the modelling of the optical elements (mirrors, filters, beam-stop). A GUI (Graphical User Interface) working in an MS-Windows environment has been built (Meyer et al. 2003b), which consists of one menu and different worksheets which follow a LIGA beam-line design, e.g. the source, the optics (front-end window, mirror, filters), the scanning stage (mask, resist), the development process of the irradiated resist, and the irradiation time.

**MODEL DETAILS**

The fabrication line is presented in figure 3; it consists of parallel irradiation of a batch of x substrates, their development and electroplating as well as a storage containing y samples (y/x batches). After the irradiation, the batch will be directly developed. Since the development is rather slow, the development equipment must be able to process several batches in parallel. The developed batch will be optically inspected.
The possible plating solutions are gold alloy, nickel alloy (Ni/Co). The electroplating will be made off-line; it will not be automated. One to z batches can be simultaneously electroplated. If the thickness tolerances are too demanding for the electroplating process, the parts will be "over-plated" and a finishing step will be introduced off-line. The exposure and development steps will also be used for the resist stripping. The quality of the metal parts will be visually checked to conform to quality standards and some statistical number of substrates will be fully checked to verify these standards. All parts adhering to specifications will be sent to the customer who will detach the parts by etching the sacrificial copper layer. This will give the widest range of freedom to handle the parts according to the customer’s requirements.

The tool processing part of the model consists of spreadsheets describing the wafer process during fabrication. We have developed the software using Borland C++ Builder Professional v6.0. The following parameters are currently taken into account:

- the synchrotron schedule (see figure 4) is included in the simulation; the synchrotron parameters (current, lifetime, beamdump) are based on statistics for 2005; the current after injection is 180 mA and the life time 15 h (see figure 5). The irradiation parameter is the exposure dose (function of the resist thickness) and the scanning distance (function of the resist area).

- development are carried out directly after irradiation. The parameters are bath(s) number (number of samples which can be developed in parallel), and the development time. After development, the samples are inspected; the parameter is the time for one inspection.

- electroplating is performed off-line. Parameters are the number of suppliers, and for each of them their capacity (number of sample which could run in parallel) and the duration of the electroplating step.

- finishing is also carried out off-line. Parameters are the number of suppliers and for each of them their capacity (number of samples which can run in parallel) and the duration of the stripping step.

- the sample is rerouted again into line for a second irradiation and stripping step, it has the priority over samples to be irradiated for the first time.

- buffers exist at the end of the line and in each supplier; the parameter is the time between forwarding and receiving batch.

SIMULATION

The results we show here are based on the following parameters:

- the yield rate is 100%
- there are always prepared substrate available
- the irradiation of only one sample at the same time is possible

Irradiation and development

Different scenarios have been tested. Different solutions are possible:

- 1. No sample in the scanner during the injection time, the irradiation of a new sample begins only when a development bath is free
- 2. No sample in the scanner during the injection time, the irradiation of a new sample ends when a development bath is free.
- 3 and 4. These cases are based on cases 1 and 2; sample could wait in the scanner during injection.

Some results are presented Table 2; the scanning distance (28mm or 90mm) plays an important role for the choice of the scenario to be employed.
Table 2: Number of irradiated (exposure dose is 4000 mA.min/cm) and developed wafer for the period 01/01/2006 to 01/03/2006; these prices concerned only the irradiation and is based on a price of 150€/per hour of irradiation.

<table>
<thead>
<tr>
<th>Case</th>
<th>500 mA.min/cm</th>
<th>1500 mA.min/cm</th>
<th>3000 mA.min/cm</th>
</tr>
</thead>
<tbody>
<tr>
<td>(1)</td>
<td>375</td>
<td>150</td>
<td>75</td>
</tr>
<tr>
<td>(2)</td>
<td>500</td>
<td>175</td>
<td>75</td>
</tr>
<tr>
<td>(3)</td>
<td>549</td>
<td>169</td>
<td>75</td>
</tr>
</tbody>
</table>

Table 3: Number wafers produced for the period 01/01/06 to 10/03/06
(1) batch:25; 25 development baths (6h; 38h; 48h); 1 supplier with 25 electroplating baths (70h; 120h; 170h); 1 supplier with 1 finishing device (2h; 4h; 5h); stripping: 400; 1000; 2000 mA.min/cm, 25 stripping baths (3h; 10h; 18h).
(2) batch:25; 25 development baths (6h; 38h; 48h); 2 suppliers with 25 electroplating baths each (70h; 120h; 170h); 1 supplier with 1 finishing device (2h; 4h; 5h); stripping: 400; 1000; 2000 mA.min/cm, 25 stripping baths (3h; 10h; 18h).
(3) batch:25; 25 development baths (6h; 38h; 48h); 2 suppliers with 25 electroplating baths each (70h; 120h; 170h); 1 supplier with 5 finishing devices (2h; 4h; 5h); stripping: 400; 1000; 2000 mA.min/cm, 25 stripping baths (3h; 10h; 18h).

**Production**

We chose scenario number 3; it is the easiest to install in the line (no computational system is needed). Table 3 and figure 5, results of the complete fabrication process using scenario number 3 are given.

The obtained results show the complexity of the fabrication line; the number of parameters being important and interdependent. For example in case 3 (dose expenditure of 1500 mA.min/cm), increasing of the finishing capacity by a factor of 5 has no influence on the wafers produced for the period shown; no increase in capacity is necessary. A similar result is seen concerning the increasing of electroplating capacity in case 2 (3000 mA.min/cm).

**CONCLUSIONS**

We may conclude that this simulation tool is powerful for the modelling of the LIGA production process. The results show the high complexity of the process, but will provide the manager a better understanding and knowledge of the system. The decision to increase the throughput of the line and to decrease cost could be influenced using this pro-
HEALTH CARE LOGISTICS
Business Process Management and Simulating of a Generic Health care Unit: Application on New Hospital of Estaing

Michael Comelli, Pierre Fénies, Michel Gourgand
LIMOS UMR CNRS 6158
Campus des Cézeaux 63 177 Aubière cedex
France
email: {comelli, fenies, gourgand}@isima.fr

KEYWORDS
Simulation, Generic Healthcare Unit, 3D emulation, Hospital Simulation, business process modelling, Business process simulating.

ABSTRACT
A modelling of a common department called “Generic Healthcare Unit” (GHU) is proposed in order to reduce development time of the decision-making tools for Healthcare.

INTRODUCTION
The New Hospital of Estaing (NHE) which will be opened at the end of 2009 will be established by the move of the current “Hôtel Dieu”, a part of the Clermont-Ferrand hospital. Several independent surgical units will be linked in one in the NHE. The objective of this change is the passage from a structure divided up to an independent unit system registered in a pool which reaches the same objectives centred on patient. Design, study and management of hospital systems are increasingly difficult and expansive as existing multiple and complex system. In order to build the future NHE decision making tools, we use a methodology called ASDI (Analysis, Specification, Design, and Implementation) (Chabrol et al., 2006). In this paper, we would like to present a small part of the global projet of modeling and simulation flow in NHE. The object of NHE modeling is to simulate business process for the future NHE. Hospital managers required specific study and model for each medical department. The idea of a Generic Healthcare Unit (GHU) which gathers all common elements of healthcare units is appeared. The formalised need is to get a generic tool allowing the study of every medical department thanks to a specification in order to study and visualise the working of these future departments. Our paper is organized as follows:
- Section 1 describes the scope of the study.
- Section 2 presents the BPM study which was done for the future GHU.
- Section 3 presents the action model building for GHU.
- Section 4 shows the first results of the implementation of GHU components on the digestive pool by simulation.

GLOBAL MODELLING PROBLEM AND SCOPE OF THE STUDY
Three approaches (Chabrol et al., 2006) allow characterizing a modeling approach by flows thanks to 3 types of modeling: macroscopic, mesoscopic, and microscopic modeling. Macroscopic modeling considers the flow in a complex system as an aggregated phenomenon, whereas microscopic modeling considers individual interactions. Mesoscopic approach incorporates entities in pack-age forms and constitutes an intermediate level between the macroscopic one and the microscopic one. Table 1 shows the coupling between these three approaches with various temporal horizons. A literature review about the GHU is given in (Fenies and Tchernev, 2005). Moreover, few papers deal with tasks conflicts. In some of them, (Fenies and Tchernev, 2005), priority tasks are taken into consideration. But there are only two types of tasks: the urgent (unpredictable) ones and the expected ones. Judging from the literature, very few papers propose a simulation where medical member choices and moves are modelled. Most of them focus on patient route and then deducted workload resources. In these papers, medical resources are considered fixed contrary to our model. Indeed, in our system, medical staff moves represent a major part of total workload.

Table 1: The coupling temporal horizons and surrounding areas of modelling ;the GHU Scope.

(Chabrol et al., 2006) proposes a methodological approach for process evaluation in health care system. This methodology allows conceiving a software environment which is an integrated set of tools and methods organized in order to model and evaluate complex health care system as a Supply Chain. The proposed methodology is applied in New Hospital of Estaing, thanks to a modeling environment (figure 1) which is called ASDI-HSC (Analysis, Specification, Design and Implementation for Health care Supply Chain). Figure 2 presents the modeling process which is followed by teams of the "modeling workshop in order to create and implement a generic health care unit for the NHE decision making tool. ASDI-HSC includes :
- a system of performance evaluation, core of the environment; the performance evaluation layer allows the development of one or more models of action according to the objectives of modeling. SIMAN V and WITNESS are indifferently used as core of this software environment. PREVA approach (Chabrol and al., 2006) thus allows to the constitution of action models whose objectives are centered on the creation of value, but also on "patient" satisfaction.
- a statistical part, which allows the analysis and the processing the existing data (forecasts of load, learning curves) as well as the study of the results obtained by the core of the environment.

- Graphics tools and 3D Emulation which makes possible to animate in 3D with MANTRA 4D the simulation models. This three-dimensional chart tool is very significant to imply the actors in the organisational change.

- a part which contains methods of analysis and specification which is constituted here of the ARIS and UML tools and formalisms).

- a Modelling Methodology of the field which was presented in (Chabrol et al, 2006).

One of the important aspects for behaviours orientation is to allow an 3D visualization of results obtained (figure 2) using 2D models and retranscribe in various states of results (scorecards). 3D visualization makes it possible to convince auxiliary nursing staff of solution feasibility, especially when they were implied in its modeling. Modeling process result is doubt: decision-making aid tool is legitimated and accepted (because produced with auxiliary nursing staff and for auxiliary nursing staff) and change is accepted ASD-HSC modeling environment is applied in NHE in order to conceive a GHU. This GHU is therefore implemented on NHE Health Care Unit.

**BUSINESS PROCESS MODELLING FOR THE GHU**

The used systemic decomposition for BMP (thanks to ASDI) arises in the following form, for a GHU which is integrated in the Health Care Supply Chain, as for the Health care Supply Chain system: (Details are given in Chabrol et al., 2006): (i) the Logical Sub System (LSS) contains the transactions (flows) treated by the system; (ii) the Physical Sub System (PSS), which is structured in small units (Business Unit) containing physical entities needed for elementary operations; (iii) the Decision Sub System (DSS), which is structured in decision center. The three sub-systems are thus complementary and communicating two to two. Health Care Supply Chain is considered as a finished number of Health care Units which are structured with the systemic decomposition suggested by ASDI. From macroscopic, mesoscopic and microscopic approaches organizing structure, we ordered current system’s structure according to 8 descriptive levels (figure 3). To collect information on actual hospital is a need to conceive future works of care units. To reach this goal, a research protocol consisted of two qualitative studies undertaken in parallel was done:

- using semi directive qualitative individual interviews (56 interviews) with the whole of departments’ heads of actual Hotel Dieu and health executives, a first study made it possible to position and determine processes crossing health care unit in HSC. Positioning results of care unit in S of actual hospital were already validated to some extent by medical personnel and actors taking part in hospital processes (Chauvet and al., 2005).

- using groups interviews (20 groups) with actors of future digestive pole, a second study must make it possible to make emerge knowledge model of care unit by analyzing similarities between various care units which will cohabit in future organization (Fenies and Tchernev, 2005).

These tasks have been defined for a journey of an actor for each service. All the actors concerned by d representation have been nurse, assistant care and agent of hospital services. Elementary functions need a more precise decomposition to describe business process for GHU. This decomposition is represented with functions trees. Thanks to this knowledge collect, we did try to identify generic processes which have been translated in software components for simulation. This part of the modeling process is given in next section.
ACTION MODEL SPECIFICATION

The first paragraph of this section describe the tasks management specification and the second paragraph gives the human move specification.

Tasks management specification

Simulating the medical staff work is a very complex task, since the work for each medical people depends on series of executed tasks which involve other ones. Whereas it is possible to distinguish fixed activities like food round, medical care round, many others tasks are caused by stochastic events e.g. patient call. Indeed, the work executed by a medical member is often created by a choice between several tasks. This choice involves an automated tasks allocation among medical staff. LSS and PSS allow this understanding. In order to build the best model as possible, tasks conflict management in real time is necessary. Thus, the simulation model has to be able to establish priorities for each task and especially to stop medical member achieving a task in order to allocate him another one more priority.

This thought induces simulation experts to conceive a task management module where goals are as follows:

- Update and creation of new tasks
- Priority task definition
- Tasks allocation

The function “Update and creation of new tasks” means that this module has to update or create others tasks according to the LSS and the DSS. For example, if a patient calls a nurse, the activity “see patient in room X” must be generated.

The functions “Priority task definition” and “Tasks allocation” consist in defining which tasks are priority and which medical members have to achieve them.

The proposed model is composed of two modules, the first one performs tasks management as mentioned, and the second one performs tasks simulation. The figure 4 describes links between both modules and show where knowledge model is used.

Human move specification

The last simulation concept developed to simulate medical department work is the people move. A module which computes the shortest route between two places has been developed. It is not specific to our study but also developed for other simulation studies of this project. Indeed, if we consider the PSS, there are about forty places where people can move.

Figure 3: Descriptive levels of Business Process Modelling for NHE Knowledge model

Figure 4: Simulation model

In our case, structure constraints are corridors. Even if the shortest route problem is not NP-difficult, the implementation of such a module is not an easy task.

To compute routes, two solutions exist. The first one consists in computing all routes before the simulation. This one can be implemented for small size. Otherwise, computations are made during the simulation and are written to memory.

Action model implementation

A major element of this model is this ability to pre-empt a medical member at any time and this ability to consider any event at any time in order to be able to run the tasks management module. More precisely, “medical member pre-emption” means that the simulation model has to be able to stop medical member during a task and to allocate him a new task.

So, in order to achieve this goal, activity based simulation has been preferred to event discrete simulation, since it allows to simulate pre-emption and event consideration in real time thanks to small buckets. In fact, at each bucket, tasks management module is run, what means tasks priority and tasks allocation are deduced. Then, in the same time, the medical member activities are changed or not.

The simulation model has been developed with Witness 2005. A bucket size of 1 second is used and seems to be a maximum time to take into account new events.
APPLICATION OF THE GENERIC HEALTH CARE UNIT COMPONENTS IN NHE ON THE FUTURE NHE DIGESTIVE POOL

The generic health care components which were developed for the GHU have been applied in digestive unit of future NHE. The first paragraph of this section gives the simulation hypothesis, the second paragraph the simulation results.

Simulation hypothesis
Despite the proposed simulation model, many simulation hypothesis have to be defined. The model validity depends on the relevance of these ones. This healthcare unit is constituted of 20 rooms, 1 office room, 2 secretariats, 1 crew lounge, 2 workrooms, 1 milling room (Figure 5). The day shift is composed of 3 nurses called N1, N2, N3 and 2 orderlies O1 and O2. N4 and N5 belong to night shift. N1, N2, N3 and N4 arrive at 6:00 am whereas O2 arrives at 7:00 am.
Thus, medical staff takes part in this study and validates the collected data. Simulation hypothesis can be divided into four parts as follows:

Medical staff
- Medical staff working hours are not fixed, and overtimes can be used in case of unexecuted tasks.
- Unit is divided into sectors.

Patient
- Patient allocation is made according to a smooth algorithm.
- Patient call is modelled thanks to Poisson distribution.
- A single pathology is assigned to each patient.
- Daily patient arrival is modelled thanks to Gauss distribution.

Task
- Multi-resources tasks cannot be pre-empted.
- Medical member process time is considered identical for multi-resources tasks.
- Different tasks can be achieved in the same place at the same time.
- Medical material is not considered as capacity constraint.
- Task time process is modelled thanks to Gauss distribution.

External system
- Logistic and human flows which cross medical departments are not simulated but are taken into account by the tasks management.

The building of such a simulation model requires many programming time. Thus, we present a first study which allowed to validate simulation approach thanks to analytical results and 3D emulation. This one consisted in simulating the two first hours of the day shift of a “Hotel Dieu” healthcare unit.

The succession of their general tasks is known and given by Graph 1. The number and the duration of this tasks depend on the number of patients in the healthcare unit. This is achieved by the task management module which define all the tasks.

Table 2 presents a sample of tasks generated by this module. For each activity, we know who and who will achieve it. The term “Beginning” of this table means from when the task can be performed. The precedence constraints are taken into account by the priority task management. A task cannot be performed before all the more priority tasks are achieved.

Table 2: Generated Tasks

The purpose of this first study is to evaluate the workload of medical staff according to three levels of bed occupancy (100%, 75%, 50%). We do not consider stochastic events (patient call,...) and consider all the durations as deterministic.

We use the experimental framework of Orên which is divided into five parts as follows:
Objective: Estimating the medical staff workload according to different bed occupancy.
Input variables
- 3 levels of bed occupancy (100%, 75%, 50%)
Output variables to observe
- Work time of Medical Staff during this period
- Moving time of Medical Staff during this period
- Unactivity time of Medical Staff during this period

**Finishing conditions of the simulation**
- End of the studied period (7:30 a.m)

**Initial state of the system**
- State of healthcare unit at 6:30 a.m, i.e., night shift = 2 nurses

**Results presentation**
- 1 replication
- Graph and table analysis
- 3D emulation

**Results**
As mentioned by the experimental framework we observe the three variables as follows:
- Work time of medical staff during this period
- Moving time of medical staff during this period
- Unactivity time of medical staff during this period

The work time stands for the amount of time spent by medical members in office room, feed room or patient room. The unactivity time stands for the amount of time spent by medical members in the crew lounge. The moving time stands for the amount of time spent by medical members in the corridors.

Judging from Graph 1, Graph 2 and Graph 3, bed occupancy is a factor which influences largely medical staff workload.

What seems more relevant is the evaluation of moving time which is very different for each medical member. This can be explained by the location of major offices which are remote. The fact that the unactivity for each member is positive even for a bed occupancy of 100% do not correspond to the reality lived in HD. This can be explained by the fact that stochastic events are not considered.

As mentioned, the second purpose of this study is to visualize medical staff work and more precisely department activity (figure 6). Many previous simulation hypothesis have been defined in order to achieve this goal. For instance, medical staff move was an important development constraint. Nevertheless, these constraints ensure a detailed model. We use the "MANTRA 4D" software to build the 3D model.

**CONCLUSION**
In this paper, we have used ASDI-HSC, which is a modeling environment in order to conceive a Generic Health Care Unit and to apply it on NHE. The proposed environment, which is a bridge between computer sciences, management sciences and operational research, gives a solution to conceive decision making tools which can also be used in meeting in order to manage changes.

The proposed model has been shown at a part of medical staff in order to show how "modelling workshop" uses collected data. Our model was used to reduce medical change resistance. Model presentations induce a lot of questions from medical staff about its future work.

Therefore, the extension of discrete event simulation as a tool for changes management will open a new research field between computer sciences and management sciences.

**References**


MODELLING TEMPORAL ASPECTS OF CLINICAL OBJECTS

Peter SUMMONS
1 The University of Newcastle
University Drive
Callaghan, NSW 2308, Australia
Peter.Summons@newcastle.edu.au

Joël COLLOC
University of Le Havre, 25, rue Philippe Lebon
BP 540 F-76058, Le Havre, France
LEACM, EA654, ISH, University of Lyon 2
16 , Avenue Berthelot 69363 Lyon Cedex 07 France
Joel.colloc@univ-lehavre.fr

KEYWORDS
Object-oriented, O-O model, time, medical diagnosis, decision support, clinical object.

ABSTRACT
This paper describes some temporal constructs that can be used in modelling temporal relationships of objects involved in medical scenarios. The paper proposes that the temporal constructs are useful and can be applied to general Object-Oriented (O-O) models used in clinical applications.

INTRODUCTION
These temporal constructs presented in this paper were developed for a Medical Decision Support (MDS) system that has been used for knowledge acquisition in a clinical setting, involving the construction of knowledge models of clinical experts’ interpretations of clinical guidelines for Patient Controlled Analgesia (Summons, 2005).

The constructs are based on the interval algebra of James Allen (Allen, 1984), which has been modified to allow time-point references to be modelled pragmatically as intervals, and also extended to represent the needs of a pragmatic medical model created by the MDS for specific medical domains.

1. TEMPORAL RELATIONSHIPS IN MEDICINE

The timeliness of a medical decision, or of the data required to support the decision, is of course relative to the context of the problem for which a support tool is designed. In some medical domains the temporal context of a decision may be expressed in days or months, and reference may need to be made to previous decisions covering longer time spans. Such cases require very expressive and big-picture views, for example when expressing the concept of recurring, which is required to identify and describe large-cycle diseases such as a chronic gastric ulcer (Colloc and Summons, 2003).

The medical domain uses relative temporal references extensively. An adequate temporal framework requires a rich representation of time-points, intervals and mechanisms that support absolute and relative references. Temporal ordering between these also needs to be supported. As an example, the medical task of diagnosis is used to illustrate the justification for, and the variety of, temporal constructs in the medical domain.

The diagnosis of a disease is often aided by observing the presence of characteristic symptoms, such as the diagnosis of measles normally needing the presence of spots. In addition to noting the presence of a symptom and qualities of its magnitude or effect, such as the severity of the symptom, it is often required to associate temporal information with the symptom, such as the time of its occurrence, or its duration. For example, when diagnosing a patient as being unrousable, a minimum measure of the duration over which the patient is unresponsive is necessary.

Diagnosis may also require a complex description of the temporal sequence of presenting symptoms used in the diagnosis, such as in the determination of the type and stage of Hepatitis B infection, where both the order of occurrence of the symptoms, as well as their duration, have to be considered (Gammer and Nejdl 1995; Shahar, 2000).

2. TIME INTERVALS

Medical scenarios are often expressed in temporal terms relating to underlying intervals, such as peri-natal or infections. Some medical temporal references, such as trimester, involve a reference relative to a temporal point that is also part of the terminology of the domain.
In this case the domain is pregnancy and the reference of trimester is to the temporal domain “point” conception. Other medical temporal references involve purely relative references to the inter-interval relationships that may exist, while still others involve a combination of the two.

As seen above, the establishment of time-points to anchor an interval relationship is also very important in many medical scenarios, such as when conception is used to anchor intervals in a pregnancy condition. Knowledge of the conception date is necessary in the determination of whether, and how far, a fetus is in its first trimester. For example this might be needed both from a legal and a medically efficiency viewpoint, for decisions regarding removal of a fetus from its mother (abortion or delivery).

Medical reasoning uses temporal interval representations in both an absolute and a relative manner. As indicated earlier, both the ordering of interval occurrence, as well as the duration of the intervals, are important in the description of the stages of some diseases or in the monitoring of their progress. However, different levels of aggregation and details must be taken into account to model the evolution of a disease. Each stage of the disease evolves with a specific scale of time, some of which are very fast, while others are very slow. These episodes can also be nested (Colloc and Summons, 2003).

2.1. Natural Temporal Representation

The most formal of the temporal modelling approaches use intervals as their descriptors, with their reasoning based on Allen’s interval algebra (Allen, 1984). Intervals, based on Allen’s temporal relations cater for complex medical abstractions (Larizza et al, 1997).

Advantages of schemes employing intervals occur in the reduction of system storage requirements and the capacity to retain a “big picture” description of medical scenarios where episodic attribute values are aggregated into event dependant states based on the state’s interval corresponding to the duration of the state. This allows temporal interpretations that would be difficult if attributes (possibly with many repeated values) at discrete time-points were used to describe a state. Temporal abstraction is a requirement of clinical guidelines so that they can identify and represent long-term evaluation of a treatment and long-term repeated patterns, and also short-term detection of changes in a patient’s condition (Seyfang and Miksch 2004). In the determination of a gastric ulcer, a medical scenario requires the big picture recognition of reoccurring conditions so that the reoccurring periodic cycles can be interpreted from the shorter manifestations. This requires the ability to express the occurrence of generalised and abstracted states in terms of intervals.

Users and medical domain experts can express events and durations in natural terminology represented by interval representations. Intervals can be used to meaningfully interpret statements such as “before the contraction”, “during the period of epigastric pain”, or to meaningfully refer to the “duration of a headache”.

2.1.2. Context Dependency of Time

Another requirement of temporal relationships in the medical domain is their capability for use in a local context (Baker and Feder, 1997). The definition of an interval may vary depending upon location, for example when different medical teams have established different time-points for what is considered the safe removal of a fetus, or may need to adhere to local legislation determining when a fetus is viably recognised as a child. Also, the temporal definition may have to be interpreted within the context of other domains, such as the legal domain of the fetus removal example. These depend upon the region under consideration (country, county, state, etc) as the legal ontology has precise local definitions and specific temporal interpretations for situations such as abortion.

3. MINIMUM INTERVAL CONCEPT

The temporal logic presented in this paper is based on that of Allen, as the temporal interval relations are simple and correspond to realistic usage in the medical domain. As discussed earlier the medical domain ontology has varied temporal description requirements, covering relative as well as grounded temporal relationships. Intervals are needed to describe episodic features, including episodes in which the actual duration or absolute time references may be unknown. Allen’s interval logic did not explicitly recognise a time point (instead referring to an analogous interval called a moment), however time point references are necessary in the medical domain, as some knowledge is expressed in terms of an absolute temporal dimension such as the real-time scale. Such grounding points and grounded periods (intervals) can be, but are not necessarily, absolute in the sense that they are required to be anchored to a specific time dimension. They are simply points or intervals to which other points or intervals are referenced.

We introduce a temporal concept of a minimum interval duration. This is similar to Allen’s moment in that it cannot have a defined sub-interval, unless that sub-interval is equal to it, but it differs from the definition of a moment in that it does possess duration as a measurable quantity. It is representative of the smallest duration interval that would need to be referred to in the domain.

The concept of the minimum interval is dependent on the domain of interest and is applied implicitly in most pragmatic reasoning systems. A medical example is
found in radiography where ultrasounds can be measured at higher definitions than those legally allowed for result interpretation. This is a pragmatic constraint imposed due to legal liability.

We adopt the Minimum Change Principle and propose that the state of a property which holds over an interval will be maintained unless explicitly recognised as having changed. We further propose that Object states have an initial value of Normal in most medical contexts unless explicitly set otherwise. By this we mean that a condition is assumed not to be present until it is recognised, or there is a reason to suspect it is present. Clinicians will not examine for a condition unless they have a reason to.

Following from the assumption of the minimum change principle and the definition of the minimum duration interval, any property that an object possesses is assumed to persist for at least a length of time equal to a minimum duration interval. It is not possible in the domain to recognise or measure a change in the property within this interval. Similarly if an object is in a specific state, then that object state is assumed to hold for a period of time at least equal to a minimum duration interval. The concept of a minimum interval could be extended to allow for different minimum interval durations for different attributes.

Any given interval has a finite duration which can be expressed as a multiple of minimum interval durations, that is, \( n \cdot \text{min}_{	ext{dur}} \), where \( n \in \mathbb{N} \) (\( n \) is the set of positive integers excluding 0). This means the Dividing Instant problem, originally posed by Bentham (Bentham, 1983), cannot occur. This problem involves a decision as to whether a proposition \( p \) holds at the point at which two contiguous intervals \( i_1 \) and \( i_2 \) meet, where \( p \) is held true in the earlier interval, and false in the later. The problem only occurs if the interval is believed to include its end points, treating them as time-points, in which one time point is common to both intervals. There are, of course, interpretations in which both an interval’s end-points, or particular interval end-points (either start or finish), are excluded from intervals. These approaches are, however, described as being unjustifiable and artificial (Ma and Knight, 2001). It does not occur in our proposal as there can always be found a finite interval over which a recorded state or an initially assumed normal state is constant.

We do not allow intervals to be continuously decomposable and impose a minimum duration requirement for intervals and thus may be susceptible to the Intermingling Problem, which proposes the possibility of a property changing its truth-value infinitely within a finite duration time interval, in a similar fashion to how Allen’s moments were criticised by Ma and Knight (2001). However, the pragmatic assumptions of the minimum duration interval, that these are the smallest intervals over which change can be measured or observed in a specific domain, and the minimum change principle implicitly proposing that the state of a property, once established as holding in a minimum duration interval, cannot then change within that minimum duration interval, effectively eliminates any chance of the Intermingling Problem occurring.

Allen defined a Property, which can exist throughout, or persist over, an interval by means of the \( \text{holds} \) predicate. Allen’s axiom H.1 and H.2 (Allen, 1984) define the \( \text{holds} \) predicate as:

H.1  \( \text{holds}(p, i) \Leftrightarrow \forall i'(\text{in}(i', i) \Rightarrow \text{holds}(p, i')) \)

Where \( \text{in}(i', i) \Leftrightarrow \text{starts}(i', i) \lor \text{during}(i', i) \lor \text{finishes}(i', i) \)

H.2  \( \text{holds}(p, i) \Leftrightarrow i'(\text{in}(i', i) \Rightarrow i'(\text{in}(i', i') \land \text{holds}(p, i'))) \)

The \( \text{holds} \) predicate defines the continuity of a property \( p \) during a time interval \( i \), so \( \text{holds}(p, i) \) indicates that the property \( p \) holds over the interval \( i \), and that there is no sub-interval \( i' \) of \( i \) over which \( p \) does not hold. For example, if the property \( p \) was “headache” and the interval \( i \) was “the last two hours” (a temporal reference that is relative to the anchor point “now”), then \( \text{holds} \) (“headache”, “the last two hours”) indicates that a headache was experienced continuously over the last two hours, that is, at every sub-interval during the last two hours.

Although Allen’s axioms are still applicable, we modify them by re-defining the predicate \( \text{in}(i) \) in H.1. The redefinition, given in ALLEN-1 below, allows equality of the two intervals participating in Allen’s original \( \text{in}(i) \) relationship, when the sub-interval is a minimum duration interval. This means that a minimum duration interval can be its own sub-interval.

ALLEN-1  \( \text{in}(i', i) \Leftrightarrow \text{starts}(i', i) \lor \text{during}(i', i) \lor \text{finishes}(i', i) \lor \text{equals}(i', i) \)

Axiom OP-1 below is a restatement of Allen’s basic definition that the property \( P \) holds over the interval, with the extension that the interval is of minimum duration, and with the redefinition of the \( \text{in}(i) \) predicate for the MDS, given by ALLEN-1

OP-1:  \( \forall i, \forall P: \text{holds}(P, i) \Leftrightarrow \exists \delta_i((\text{during}(\delta_i, i) \lor \text{starts}(\delta_i, i) \lor \text{finishes}(\delta_i, i) \lor \text{equals}(\delta_i, i)) \Rightarrow \text{holds}(P, \delta_i)) \)
4. APPLICATION ODF TEMPORAL PROPERTIES TO OBJECTS

We extend Allen’s Axiom H.1 to express predicates that apply to objects with temporal extent and these extensions for objects are expressed in OP-2 to OP-5 below. Allen’s holds predicate defines that the property P can hold over interval i. When we talk of an object Obj having a property P, denoted P(Obj), then the holds predicate in OP-1 defines the capacity of an object to have a property over an interval. The object’s class has to be capable of having the property over the interval. A stronger predicate than holds is the predicate exists in OP-2 below, which states that if the object Obj actually has the property P over the interval i, then the object Obj exists over that interval (assuming that there is not a property such as not-exists, of course).

OP-2: \( \forall \text{Obj}, \forall P, \forall i: \text{exists}(\text{Obj}, P, i) \)

\[ \text{\iff exists}(\text{Obj}, P, i) \]

\[ \Rightarrow \text{The Object Obj exists with Property P over the interval i} \]

OP-2 extends Allen’s definition of a property P holding over a time interval i to an object existing over a time interval. In the MDS architecture, the knowledge that an instantiation of an object exists over a time interval i in a system is implied if it can be established that a property of the object is held over that interval. The object has to exist for the property to hold.

The definition of OP-2 is extended by OP-3 to cater for the existence of an object over one minimum duration interval, termed an occurrence of the object. OP-3 also shows the translation from Allen’s occur predicate (Section Erreur! Source du renvoi introuvable.), defined in terms of properties and intervals, to the occurs predicate defined in terms of Objects and finite duration intervals used by the MDS architecture.

OP-3: \( \forall \text{Obj}, \forall \delta_s, \exists S: \text{exists}(\text{Obj}, S, \delta_s) \)

\[ \Rightarrow \text{occurs}(\text{Obj}, \delta_s) \]

OP-3 indicates the minimum condition that must be satisfied for an object to be observed to exist within the domain. It uses a minimum-duration interval \( \delta_s \), the smallest time period that can be meaningfully measured within the domain. So we can say that the minimum duration interval defines the smallest duration over which an object can exist within the domain. The use of the predicate occurs is illustrated when a clinical intervention, such as a test or measurement, occurs and indicates the possibility of the existence of a clinical condition such as a disease.

The requirements for specialised existence of objects are extended in OP-4, where a definition for object dependency is established.

OP-4: \( \forall \text{Obj}_1, \forall \text{Obj}_2: \text{dependent}({\text{Obj}}_1, {\text{Obj}}_2) \)

\[ \Rightarrow \forall i, \forall P: (\text{exists}({\text{Obj}}_1, P, i) \]

\[ \Rightarrow \exists S: \text{exists}({\text{Obj}}_2, S, i) \]

\[ \land \forall S: (\neg \text{exists}({\text{Obj}}_2, S, i) \]

\[ \Rightarrow \neg \text{exists}({\text{Obj}}_1, P, i) \]

OP-4 states that the existence of an object (Obj1) is dependent on the existence of another object (Obj2). For this relationship, the existence of Obj2 is a necessary condition for object Obj1, to exist. This is a domain specific relationship and, as such, a pragmatic one. For example, it will usually occur between objects participating in is-part-of relationships, such as a heart is-part-of a person. Choosing a different scope for the domain may eliminate the relationship.

The dependent relationship can also sometimes occur between objects participating in is-a relationships, such as the registered nurse class being dependent on a clinical staff class (a registered nurse is-a clinical staff member). In some situations the domain may be restricted to only nursing staff and so the relationship may not be necessary in the new domain.

OP-5: \( \forall \text{attribute}, \forall \text{Obj}, \forall i, \forall \text{value}: \)

\[ \text{dependent}({\text{attribute}}, {\text{Obj}}, i) \land \]

\[ \text{exists}({\text{attribute}}, \text{value}, i) \]

\[ \Rightarrow \text{has-value}({\text{Obj}}, \text{attribute}, \text{value}, i) \]

The predicates relating to objects that are expressed by OP-1 to OP-5 rely on the Exclusion Axiom EX-1 below. It should be noted that EX-1 is based on a minimum duration interval \( \delta_i \) and on the holds predicate, which means that its result can be extended to the exists predicate, and to general intervals.

EX-1: Exclusion Axiom

\[ \forall P_i, \forall \text{Obj}, \forall \delta_i: \text{holds}(P_i(\text{Obj}), \delta_i) \]

\[ \Rightarrow \forall P, \forall \delta_i: \neg \text{holds}(P(\text{Obj}), \delta_i) \]

The Exclusion Axiom EX-1 states that if an object exists in one state value (Section Erreur! Source du renvoi introuvable.) that it cannot simultaneously exist with a contradictory state value, assuming that there is a mutually exclusive collection of all the possible state values for the object class. As mentioned in Section Erreur! Source du renvoi introuvable., the assumption of crisp set values is adopted for the sake of simplicity in this thesis. However, Axiom EX-1 precludes the possibility of representing object state values by Fuzzy sets, where the object can have degrees of different state values simultaneously. It should be noted that in a fully detailed model Fuzzy values could be assigned by to the values by associating a Belief function (Summons, 2005).

The constructs described above illustrate the extension of abstract time intervals to their association as temporal properties of objects. These have been used with other
constructs that model the occurrence of events and causality in clinical situations (Summons, 2005).

CONCLUSIONS

We have presented some constructs that can model temporal relationships of Objects in a medical context. Although they were developed for a system that has been applied to modeling knowledge in clinical guidelines, they have been presented as being topologically and architecturally independent, indicating that they may be applicable to other medical models based on the Object-Oriented paradigm, such as that proposed by Colloc and Boulanger (1993).

REFERENCES


BIOGRAPHY

PETER SUMMONS was a Systems Engineer in industry. He has a Graduate Diploma in Education. Since 1986 he lectured at The University of Newcastle Australia, where currently he is a member of the Faculty of Science and IT.
He holds a Ph.D in computer sciences by the University of Newcastle, Australia. His main research interests are health informatics, DSS and system design.

JOEL COLLOC holds a Ph.D. (1990) in computer sciences by the National Institute of Applied Sciences (INSA) of Lyon and a MD (1985) by the medical faculty of the Claude Bernard University (Lyon 1) France.
He is currently professor in computer sciences in the Le Havre University and was previously associate professor in the Jean Moulin Lyon 3 University. His main research topics are DSS and knowledge bases in medicine, CBR and multi-agent systems, nervous system modelling and cognitive sciences. He is a member of the LEACM laboratory ISH, University Lyon 2.
A decisional model for the performance evaluation of the logistic process:
Application on consultation ambulatory unit of a hospital supply

Christian Alexandre¹, Pierre Fénies², Sophie Rodier³
¹AXEGE Clermont-Ferrand  
²LIMOS UMR CNRS 6158  
³Université d’Auvergne, IUP Management et Gestion des Entreprises  
France  
email: alexandre@axege.com, {fenies, rodier}@isima.fr

KEYWORDS
Advanced planning and scheduling, advanced budgeting and scheduling, hospital supply chain, performance evaluation.

ABSTRACT
We propose a generic decisional model allowing to estimate in a total way (physical flows, financial flows) plans for any system contained in supply chain. We present how to integrate this model in a software suite dedicated to supply chain and call this type of software presenting a global decisional solution Advanced Budgeting and Scheduling. To show the generic character of our approach, we apply the chaining of models suggested to the logistic process of a consultation ambulatory unit of a Hospital Supply Chain.

INTRODUCTION
Activity pricing (T2A) for the public hospitals as private is durably modifying the management of health care systems in France. This change of hospital financial system is combined with the centring on the main processes of care, involving an externalisation of logistic functions. Hospital system became a system opened on the outside which interacts with external logistic entities or medical service provider. The comparison with industrial supply chain is obvious: the current hospital, taking into account its growing complexity, is closer to an immense logistic chain whose agents aim at satisfying the patient. We propose to define the contemporary hospital as Hospital Supply Chain (HSC). Thus, a HSC is an opened unit crossed by human, material, informational and financial flows, composed of varied autonomous entities: suppliers, hospital departments (emergency, operating theatre suite...), logistics providers, medical providers... These entities use material resources in a limited number (material, capital...) and coordinate their action by an integrated logistic process in order to improve firstly their collective performance (satisfaction of the patient, optimization of the hospital system operation) but secondly, in long term, their individual performance (maximization of created value by an entity). In the centre of this definition rests the concept of process, which is a whole of activities intended to provide a product or a service which contributes to reach the objectives of the system. The process activities transform input elements into output elements while bringing an added value. The value is created by combining process in network. The value provided by an organization results from a combination of activities and process, or value chains. The complexity of a SCH, as of an industrial Supply Chain, is due to the structure of the logistic process (Theiner 1997) and to the number of entities which interact simultaneously. Also, any modelling of the SCH rests, firstly, on its logistic process modelling. The object of this paper is to propose a generic model for the processes evaluation, and to validate it on a HSC.

If many studies on Anglo-Saxon hospital systems show that Activity Based Costing (ABC) are those which conceptually present the best solution for HSC, it exists only few approaches integrating ABC in hospital systems. We propose to consider the hospital as Supply Chain and to evaluate it like such. The evaluation suggested by the ABC valorization models has only one informational value and does not allow to control an organization from a financial point of view because it does not evaluate the position of treasury, nor the creation of cash flows. However, it translates the elements of process physical flow into information which makes it possible to estimate the elements of financial flow (Fénies and Gourgand, 2005).

To control, configure or to design logistic flows of Supply Chain, Advanced Planning and Scheduling (APS) constitute software suite dedicated to Supply Chain. Thus an APS gives a total planning for the whole of Supply Chain and allows by a hierarchical approach a decomposition of planning on the whole of the chain entities on the short, medium and long term horizons. The APS integrate neither in a conceptual way nor in an operational way the financial flow constraints.

We propose to evaluate the financial flow thanks to the typology presented on figure 1.

Figure 1: Various levels of financial flows budgeting in the software of the field.

The study of the existing tools on the market, as the approaches existing in the literature, lead us to define a decisional tool allowing the evaluation of set of flows for Supply Chain. We call it Advanced Budgeting and Scheduling (ABS). We integrate in ABS a decisional module which organizes information for Supply Chain managers in the form of prospective scorecard for the ex ante evaluation of activities (Fénies et al., 2004).

The supply of the centred part on the costs and the financial part of the prospective scorecard is carried out by the model PREVA which makes it possible to translate in a prospective and causal way the impact of physical flow into element of
financial flow by a succession of models of the Activity Based Costing type. PREVA evaluate in the medium and long term cash flows generated by Supply Chain and the entities which make it up, but also in the short time the position of treasury of the entities of the Chain generated by a collaborative planning.

In a first part, we propose a generic approach for the processes evaluation which constitutes the "decisional" part of the software suite previously presented. In a second part, we propose the use of this approach on an ambulatory unit of consultation of a HSC.

**PROPOSAL FOR A GENERIC DECISIONAL MODEL**

In this section, we present the approach used to couple physical and financial flows in the processes evaluation. We call this approach PREVA for PProcess EValuation. This approach makes it possible either to evaluate various plans suggested by mathematical actions models or resulting from simulation, or to be used to build action models integrating of financial flow constraints.

**Principle of the approach**

The study of the various valorization methods (Chabrol et al., 2005) shows that the models ABC are adapted to evaluate the costs in the HSC. We suppose that Supply Chain consists of 1 to n Business Unit. We suppose as well as the indirect costs are explained by the ABC cost drivers. To use ABC to model the costs of the HSC, we propose to give for each business unit (cares departments, logistic unit, units of support...) composed of at least an elementary logistic process the various elements necessary to represent physical flow plantings into element of financial flow. we propose (Figure 2) to design and evaluate a planning in three steps.

**Supply Chain Customer Flow Evaluation**

- Quantities made
- Standby quantities
- Customer satisfaction

**Supply Chain Evaluation with ABC system**

- Thanks to resources: consumption, direct and indirect costs

**Supply Chain Global Evaluation**

- Financial metrics

**Supply Chain Financial Evaluation**

- thanks to resources: consumption and payment period

**Figure 2: Translation of the physical flow impact in element of financial flow**

Thus the sequence of the steps is as follows:

**Step 1:** an action model (simulation/optimization), designed starting from the knowledge model provides a planning which gives the quantities of products and services to be treated, possibly the services and products treatment order, the physical quantities of cost drivers (at the tactical level), even the process type (at the strategic level) but also customer satisfaction (with the logistic meaning of the term). This model of simulation is exclusively centred on the customer flow physical criteria and will feed in information to the financial evaluation modules. The decision-making of Supply Chain Manager will be modelled using heuristic able to integrate a financial dimension in the choices.

**Step 2:** our generic approach must make it possible to determine the indirect costs consumption thanks to the logistic process costing evaluation for each business unit and for the whole chain. In this context, the ABC cost of Supply Chain is the sum of the process costs in each business unit with the sum of the elements customers direct costs. The stocks value can be also evaluated on any level of the chain. The potential of value creation is also evaluated by combining the difference between the request and the quantity sold by a business unit or by the whole chain with the margin on direct costs. Between each element of the chain, the transactions are evaluated in transfers price when the processes relate to the same legal entity or in market price if the element of the chain is in contact with the final customer or businesses unit pertaining to different companies. The differences picked out between ABC costs and transfers or market prices make it possible to the managers to evaluate the value creation (level of profit) in each entity of the chain. The various cost drivers, revealed by the analysis of the processes during the development of the knowledge model (with each elementary process a cost driver is associated) allow an indirect consumption evaluation consequently resulting from collaborative planning. The figures 3 and 4 explain the selected principle to evaluate using an ABC approach for the logistic process.

**Step 3:** the direct and indirect financial resources consumption, as well as real cash balance, are given for each Business Unit of the chain. The difference between nature of the costs (calculated costs and real costs) and nature of the delay of payment for each type of resources and customers, creates a significant difference between the levels of profits and cash flows over the same period of analysis. That’s why the previous periods impact the current period for cash-flow evaluation. By the integration of the terms of payment on the level of the resources consumption, the cash position level and the cash flows are appraisable entirely for each business unit of Supply Chain like for Supply Chain.

**Formalization of the approach**

The various metrics resulting from PREVA approach can be structured in income statement form (Comelli et al. 2005). This income statement constitutes an information document which allows the decision-making.

**Figure 3: PREVA - local approach of the models ABC**
The formalization of PREVA approach proposes the coupling of an Activity Based Costing model for the logistic process with an integration of the cash-flows. Thus, the impact of the choice on physical flow in financial flows is appraisable for any customer element and any business unit composing Supply Chain.

**Decisional horizon and use of the approach**

Although using same formalization, the chaining of PREVA with an action model will be different according to the temporal horizons. At the operational level, the action model proposes a planning containing physical consumption of the resources and the quantity of production of goods and services by business unit over the period of use. These elements will make it possible to evaluate the direct and indirect resources consumption and the sales turnover over the period of planning. This step, by the intermediation of an ABC model of the logistic process, value creation and potential of creation of value are integrated. Lastly, by the integration of the terms of payments on the resources and the sales, the level of cash-flow is evaluated. For this decisional horizon, the approach makes it possible to bind and to evaluate the cash position of a business unit chain with operational scheduling and/or plannings. Figure 5 shows the connections between PREVA and the action model used for a planning at the operational level. The model can be used whatever the level of modeling.

**Cost drivers ABC consumed by the logistic process.** By the terms of payments integration, the model evaluates the activity of the logistic process compared to the level of cash-flow. Value creation, potential of value creation as level of cash-flow by business unit as for Supply Chain are evaluated. Figure 8 illustrates links between PREVA and the action model used at the tactical level for a collaborative planning.

**Decisional horizon and use of the approach**

Although using same formalization, the chaining of PREVA with an action model will be different according to the temporal horizons. At the operational level, the action model proposes a planning containing physical consumption of the resources and the quantity of production of goods and services by business unit over the period of use. These elements will make it possible to evaluate the direct and indirect resources consumption and the sales turnover over the period of planning. This step, by the intermediation of an ABC model of the logistic process, value creation and potential of creation of value are integrated. Lastly, by the integration of the terms of payments on the resources and the sales, the level of cash-flow is evaluated. For this decisional horizon, the approach makes it possible to bind and to evaluate the cash position of a business unit chain with operational scheduling and/or plannings. Figure 5 shows the connections between PREVA and the action model used for a planning at the operational level. The model can be used whatever the level of modeling.

The strategic level (figure 7) supposes that the information provided by the action model is different: indeed, this last one, on this temporal horizon level will design the logistic process and will propose a possible solution of this last one while estimating production of goods and services and physical resources consumption over the period. The ABC model estimates the value creation consequently and the potential of value creation by business unit and type of customer element. However, the model, by the the terms of payments integration, but especially by the tax integration as well as various subsidies and assistances with the investment makes it possible to evaluate at the strategic level the cashflows awaited by each Business Unit.

**APPLICATION**

The generic model presented previously was applied to a Unit of Ambulatory Consultation (UAC). A UAC is a unit of care where the patients are treated during one day and set out again then in their residence. The objective is to evaluate according to physical and financial criteria various rules of patients priority. For that, we implement the models chaining previously presented.

**Modelling and development of the action model**

The specification of the ambulatory unit of consultation was carried out starting from the conceptual model of the HSC...
with ARIS (Sheer 2002) which was selected like formalism and knowledge modeling tool of the hospital logistic process. In the case study, specificity lies in the treatment of the patients priorities.

An agenda for patient appointment date is made for each day and is elaborated by supposing a processing time of the average patient without taking account of his pathology and its severity. In addition, any patients coming from other departments must also be treated. It is then possible that a patient is not treated the right day. Pathologies differ according to their severity, their medical resources consumption and their remuneration. The objectives of the UCA being multiple (profitability, treatment rate...), the choice of a patients treatments priority rule is complex. We suppose that physical flow is composed of patient flow. Figure 8 presents a chain of patient treatment process in a UAC. A simulation model under SIMAN V Arena of this system was developed in order to evaluate in a stochastic context the best rule of patients priority management. Our simulation is not determinist and finishing. We consider six families of pathologies : P1 to P6. The patients check-in follows an exponential law of average 15 mm. The pathologies distribution and the processing time of pathologies by the medical resources is given by table 1. The criterion of pathology severity does not correspond to an emergency criterion. The financial data are "pro format" and are built starting from a regulation system for public health care system.

We evaluate five priority rules (heuristic) following:

- H1: First patient check-in, first patient treated.
- H2: The largest criterion of severity (for equivalent criterion of severity, first in, first out).
- H3: The largest financial criterion (for equivalent criterion of severity, the most interesting financial criterion).
- H4: Processing time considered on average shortest (with criterion of equivalent time, first in, first out).
- H5: The longest processing time (with criterion of equivalent time, first in, first out).

In the case of identical criterion, rule FIFO is applied. The experimental framework developed by (Orên and Zeigler 1979) makes it possible to specify our study:

- Output variables to observe:
  - the annual average number of patients treated by pathology;
  - the annual rate of patients satisfaction;
  - the annual rate of medical resources occupation.

- Input Variables:
  - Priority rules

Initial state of the system:
- the UAC resources are free and the waiting rooms are empty at the beginning of each simulated day.

Finishing conditions of the simulation:
- one day begins at 8 AM and finishes at 6 PM.

Collect results:
- a replication consists of 365 days consecutive without restoring of the "laws".
- We made 30 replicas of 365 days and determine the average value of each observable variable.

The observed variables of the patients treatment of the ambulatory unit of consultation are then used by the decisional module resulting from PREVA.

**Results**

At the end of simulation, the annual number of patients treated by pathology, the number of patients untreated by pathology, as well as the utilisation ratio of the medical resources then constitutes the input data of decisional model detailed previously. Then, we obtain the following results, given for each management rule (figures 9, 10, 11, 12). The analysis of the results (table 2) makes it possible to initially show the sensitivity of the indicators compared to the various rules but also to select the best rule.

The decisional approach gives criteria of selection which allow, for an equivalent or higher quality of "patient satisfaction" to select the most advantageous rules of management on the financial. If the results presented validate the decisional approach, the various rules of management and their order are relevant only compared to the case study.

**CONCLUSION**

The approach suggested provides more than an ABC model. Indeed, it improves the visibility of the costs but especially it shows how physical flows are transformed into financial flows. In our case study, our evaluation model is coupled with a simulation model. This choice enables us to integrate to take into account stochastic phenomena (such as the request for care variability, medical resources area...) with an aim of studying the strategies robustness. This generic approach enables us to consider, in a next study, the integration of ABC model and financial flows in mathematical models in order to improve running of HSC. We also wish to propose an extension of this approach in external Supply Chain in order to improve their functioning and to allow the managers to share the value creation realized by collaborative planning.
Table 1: Initial data

<table>
<thead>
<tr>
<th>Distribution law</th>
<th>P1</th>
<th>P2</th>
<th>P3</th>
<th>P4</th>
<th>P5</th>
<th>P6</th>
</tr>
</thead>
<tbody>
<tr>
<td>Saisonnalité of pathologies, but on average annual 1/6 simulation takes account of the saisonsnality</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Processing following a Normal law</td>
<td>N(14,2)</td>
<td>N(15,5)</td>
<td>N(21,5)</td>
<td>N(25,5)</td>
<td>N(15,5)</td>
<td>N(20,2)</td>
</tr>
<tr>
<td>Severity (of the least serious with most serious)</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>3</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>Pathology price (€)</td>
<td>20</td>
<td>17</td>
<td>22</td>
<td>30</td>
<td>15</td>
<td>25</td>
</tr>
<tr>
<td>Pathology margin (€)</td>
<td>10.62</td>
<td>8.75</td>
<td>6.5</td>
<td>7.98</td>
<td>8.8</td>
<td>8.37</td>
</tr>
</tbody>
</table>

Figure 9: Treated patients number (annual office pluralities)

Figure 10: Patient satisfaction rate (average annual)

Figure 11: Generated cash-flows (annual office pluralities)

Figure 12: Value Creation (annual office plurality)

Table 2: Selection of management rules

<table>
<thead>
<tr>
<th>Management rule</th>
<th>H2</th>
<th>H3</th>
<th>H4</th>
<th>H5</th>
</tr>
</thead>
<tbody>
<tr>
<td>A number of patients</td>
<td>sol &lt; H1</td>
<td>sol &gt; H1</td>
<td>Best rule</td>
<td>sol &lt; H1</td>
</tr>
<tr>
<td>Satisfaction Patients</td>
<td>sol &lt; H1</td>
<td>sol &gt; H1</td>
<td>Best rule</td>
<td>sol &lt; H1</td>
</tr>
<tr>
<td>Resources use</td>
<td>id H1</td>
<td>id H1</td>
<td>id H1</td>
<td>id H1</td>
</tr>
<tr>
<td>Value Creation</td>
<td>sol &lt; H1</td>
<td>Best rule</td>
<td>sol &gt; H1</td>
<td>sol &lt; H1</td>
</tr>
<tr>
<td>Generated Cash flows</td>
<td>sol &lt; H1</td>
<td>Best rule</td>
<td>sol &gt; H1</td>
<td>sol &lt; H1</td>
</tr>
<tr>
<td>Selection</td>
<td>eliminated</td>
<td>H3 and H4 are better than H1</td>
<td>eliminated</td>
<td></td>
</tr>
</tbody>
</table>

REFERENCES


Féniès P., Gourgand M., (2005), Towards an integration of physical and financial flow in software for company Supply Chain, 7th manufacturing accounting research conference in Tampere, Finland (June).


SIMULATION IN ROBOTICS
ROBUST PREDICTION AND TRACKING METHOD FOR A MOBILE ROBOT NAVIGATION

Krzysztof Skrzypczyk
Department of Automatic Control
Silesian University of Technology
Akademicka 16, 44-100, Gliwice, Poland
E-mail: krzysztof.skrzypczyk@polsl.pl

KEYWORDS
Game theory, motion planning, trajectory prediction, tracking, navigation.

ABSTRACT
The paper addresses a problem of a mobile robot navigation in case where the target is not static. Solution of such a problem requires two main issues to be taken into account. First is a prediction of a future location of the target, second is a fast tracking algorithm that is robust to an uncertainty of data. In this work a method of navigating the robot toward moving target is discussed. Fast algorithm of prediction as well as the algorithm of tracking the target based on game theoretical approach are presented. Results of simulations are presented to prove efficiency of the proposed approach.

INTRODUCTION
Mobile robot is a machine that is intended to operate and execute tasks inside a workspace, a model of which is unknown or only partially known. Moreover, very often there is an additional requirement that the robot should do its tasks in an autonomic way – that means without being supervised by a human operator. The tasks a mobile robot is considered to perform are mainly related to the problem of moving the robot from the initial location to the desired one. An application of the problem stated above, can be the task of transporting parts inside of a factory. This kind of task of moving a robot from one location to another while avoiding environmental objects is called a basic navigation task execution. It imposes serious problems on a control system. The most significant are: collision free trajectory planning, a world model synthesis and sensors uncertainty handling. The problem sketched above is a subject of very intensive research that have been conducted for last 40 years. The problem is getting more and more complicated when the target of navigation is not static. With moving target the problem becomes not trivial even when the workspace is free of obstacles (like it is in case of a static target of navigation). This kind of problem is often called a tracking one (Liao et al. 2003; Gass 2005). In practice such a problem has to be considered very often in case of designing a system dedicated to handle so called human-robot interaction tasks (Riley et al. 2000) where human is interpreted as a moving target. Solving this kind of problem requires two main issues to be taken into account: prediction of movement of the target as well as fast tracking algorithm. The importance of trajectory prediction and its influence on the quality of planning of the robot’s motion is clear. The better estimate of a future location of the target is obtained, the more effective tracking trajectory can be planned. The problem of prediction of trajectory is well represented in the literature. The most popular approach to a solution of this problem is Kalman filtering method (Åström and Wittenmark 1997; Julier and Uhlmann. 1997). Another one, often exploited in the mobile robotics, is a probabilistic approach called particle filter (Gustafsson et al. 2002). Unfortunately these approaches give only good, short term prediction of the trajectory. Moreover, an implementation of these algorithms is difficult due to their complexity. In this paper we present another approach to the problem of trajectory prediction which gives good results even when the horizon of prediction is relatively long. The method is based on using appropriate approximation tools to compose trajectories as sequences of geometric features that appear in practical mobile robot paths. That results in the method that is not an effective tool of prediction in general. The proposed approach fits well to the problem of moving object’s trajectory prediction. Thanks to a simplicity of the method it is very easy to implement and is very fast.

Another problem is related to tracking a given trajectory. If the trajectory is known the task is rather simple and there are many solutions to this problem. But when we have to track the target using only on the estimation of its location the problem is getting more complicated, especially due to the errors of the prediction method or (and) unpredictability of the target movement. Then a robust algorithm of tracking has to be applied. A good solution is the use of a strong feedback in the system. On the other hand such an approach implies that computations have to be repeated more often.

The approach to a synthesis of the robust algorithm of tracking is presented in this paper. The method discussed here utilizes elements of game theory and it models the tracking process as a zero sum game between a player (robot) and a nature (errors of prediction). Such an approach gives very good results which is proven by simulations that had been run. The results of exemplary simulations are presented and discussed in the further part of this paper.

The System Architecture Overview

Figure 1 shows a general diagram of the robot control system. Information flow in the system can be briefly described as follows. The robot is assumed to have a sensory system that is able to provide information about a location of
the tracking target. These data are stored in a target state buffer of length $M$ and are used to build a target trajectory model. Using the model the predicted location of the target is computed. Next stage of the work of the system is to plan tracking trajectory of the robot. The core of the tracking algorithm is to model the process of robot motion planning to the target location as a zero-sum game between a navigation system and a nature that is perceived as a source of uncertainty and tracking errors. The solution of the game is the base for computing the set points for the motion controller. In case when the tracking error becomes greater then some threshold value the procedure sketched above is repeated with new adapted parameters.

Now the problem can be stated as follows: Find the control of the robot that allows to reach the target in the time $t_{H} = H \Delta T \quad H = 1,2,\ldots \ldots$ . So we look for a control vector:

$$U^* = [u(t_n), u(t_{n+1}), \ldots, u(t_{n+H-1})]$$

(4)

that provides the following:

$$\|X(t_H) - P(t_H)\| \leq \varepsilon_f$$

(5)

Where the distance in (5) is measured according to euclid norm and the $\varepsilon_f$ is the accuracy of tracking.

THE METHOD DESCRIPTION

To design a method that solve the problem stated in the previous section, two issues have to be considered. First of all an algorithm of prediction of the future location of a target has to be provided. Moreover, a time horizon of the prediction has to be long enough. It is intuitively understood that the longer horizon of prediction is provided the more effective control can be performed. Next problem is to create the control algorithm that allows to reach the moving target, taking time constraints and errors of prediction method into account. Details of our approach to the solution of these two problems are presented in the following two subsections. In the last, third subsection the overall algorithm of tracking is described.

Prediction Method

The problem of prediction can be generally stated as follows: Using information about present and past state find the estimate of the state of the object in the future. There are many methods of solution of the problem, which were listed in the introduction of this paper. Unfortunately they only fit well to problems where short time prediction is needed. Additionally these methods are hard to implement as regards computation cost. A method presented in this section, though less universal than the others is free of these two disadvantages.

The method is dedicated to predict a future path of the moving object basing on the assumption that the route of the object does not change very often. This assumption covers a major number of situations (observed trajectories) known in practice. Let us first precise the problem that will be considered hereafter. Using $M+1$ observations of past and present states of the target

$$P(t_{n-1}), \ldots, P(t_{n-2}), P(t_{n-1}), P(t_n)$$

(6)

find the estimate of the state of the target in the future:

$$\hat{P}(t_H) \quad t_H = t_n + H \Delta T$$

(7)

where $H$ will be called the horizon of prediction hereafter. The idea of the proposed method of prediction can be described as follows. In the given moment of time $t_n$ using $M+1$ past observations of the state of the target (6) the model of the trajectory of the target is estimated. Here in the
paper we propose using well known polynomial regression method to find the model of a time plot of each coordinate of the target location separately. So we look for the following second order model of the trajectory:

$$\hat{P} = \begin{bmatrix} \hat{x}_t = f_s(t) \\ \hat{v}_t = f_s(t) \end{bmatrix} = \begin{bmatrix} a_1 t^2 + b_1 t + c_1 \\ a_2 t^2 + b_2 t + c_2 \end{bmatrix} t = t_{n-1}, t_{n-1-1}, \ldots \quad (8)$$

The unknown coefficients $a$, $b$ and $c$ can be computed by doing a least squares fit, which minimizes the sum of the squares of the deviations $e$ of the data $(6)$ from the model $(8)$. Using the model $(8)$ we can estimate a hypothetical location of the target in the future $(7)$. It is easy to distinguish two main issues that influence an accuracy of the prediction method stated above. First is the number $M$ of observations of a state of the target. The second is a shape of the trajectory of the target. If the target changes a direction of its movement very rapidly that means its movement is unpredictable and the method fails. Then to make the method work properly we need to use a feedback from current observation of the state of the target. So if the error of prediction:

$$e(t_k) = \left| P(t_k) - \hat{P}(t_k) \right|$$

in the moment $t_k < t_H$ is greater than some threshold value $e_T$ that means the model is no longer valid and new one has to be determined. The new model is based on another $M+1$ observations $P(t_{k-M}), \ldots, P(t_{k-1}), P(t_k), \hat{P}(t_k)$.

Tracking Algorithm

Having prediction $(8)$ of the target location the control vector $(4)$ has to be found that enable the robot reaching the target in the time $t_H$. If the prediction was perfect the problem would be rather simple. Unfortunately the prediction is not perfect and the algorithm of control has to cope with this fact. Here in the paper the method based on the theory of game (Basar and Olsder 1982) is proposed. Such an approach allows to soften effects of inaccurate prediction. The process of determining the control of the robot moving toward the target is perceived and modeled as a zero-sum game between two virtual players. The first one is the control system itself which aim is to minimize in each moment the distance of the robot to the target. The second “player” is a nature which is perceived as the source of the uncertainty and errors. The second player select these actions that provide the cost as high as possible. So goals of the players are opposite. In the rest of this section detailed description of a process of synthesis of the control law will be presented.

First of all let us define the game associated with each discrete moment of time:

$$G(t) = \{N, I, A\} \quad t = t_{n}, t_{n+1}, \ldots, t_{n+M-1} \quad (10)$$

where $N$ is a number of players (in this case $N=2$), $I$ is a cost function which determines costs associated with particular actions taken by players: $I : A \rightarrow \mathbb{R}$, in our case in the moment $t$ of time:

$$I(a_i, a_j) = f(a_i(t), a_j(t), X(t), \hat{P}(t_H))$$

where $a_i \in A_i$, $a_j \in A_j$ and $A = A_i \times A_j$ is called action space of the game. To state the model of the game $(10)$ an analytical form of the function $(11)$ has to be found and the action space of the game has to be determined. Let us start with defining an action set of the player-robot. An action of the robot is a control $(2)$ of movement of its driving platform. Both angular and linear velocities change from their minimal to maximal values. If we want to obtain discrete action set a discretization has to be done. It is easy to notice that the way of discretization influences directly the size of the problem. Here in this work we propose a method of discretization of the control space that reduces the size of the $A_j$ to the number numerically tractable. The idea is very simple – there is no need to discretize the whole control space. That is enough to make it around some characteristic values of velocities $v_o, \omega_o$ with a precision $\Delta v, \Delta \omega : A_i = \Omega \times V \quad A_j = \{ (\omega, v) : \omega \in \Omega, v \in V \}$, where

$$\Omega = \{ \omega_b - N_{\omega} \Delta \omega, \omega_b - \Delta \omega, \omega_b, \omega_b + \Delta \omega, \omega_b + N_{\omega} \Delta \omega \}$$

$$V = \{ v_o, v_o + \Delta v, v_o + 2 \Delta v, \ldots, v_o + N_v \Delta v \}$$

and $N_{\omega}, N_v$ are the number of discrete values of change of the velocities so the size of $A_j$ is equal to $A_j = (2 N_{\omega} + 1)(N_v + 1)$. The values of $v_o, \omega_o$ are determined from formulae:

$$v_o = \frac{\left\| \hat{P}(t_H) - X(t_k) \right\|}{H}, \quad \omega_o = \frac{\Theta_d(t_k) - \Theta_0}{H} \quad (12)$$

where

$$\Theta_d = \arctan \left( \frac{\hat{v}_r(t_H) - v_r(t_k)}{\hat{x}_r(t_H) - x_r(t_k)} \right)$$

The values $(12)$ correspond to the control that enable to reach the predicted location of the target $(7)$ in the time $H \Delta v$ by the shortest path. The second player symbolize nature influence that is the source of uncertainty of prediction. Since the model of discrete game is considered a finite set of possible actions have to be determined. Particular “actions” of the player are deviations of the estimation of target location from its original value $(7)$. In our approach a space around the $(7)$ is divided into a number of circular sectors inside of which an exemplary value is chosen in a random way. So the action set of the second player is:

$$A_j = \Lambda \times R \quad A_j = \{(\alpha, r) : \alpha \in \Lambda, r \in R \} \quad (13)$$

where
\[
\Lambda = \{ \alpha_i \} \quad \alpha_i = \frac{2\pi}{L} (i + \delta_\alpha - 1) \quad i = 1, 2, \ldots, K
\]  
\[
R = \{ r_j \} \quad r_j = \Delta r (j + \delta_\sigma - 1) \quad j = 1, 2, \ldots, J
\]  
(14)

The \( \delta_\alpha, \delta_\sigma \in (0,1) \) in (14) are random numbers. The size of the action set (13) is equal \( A_2 = KJ \).

The Cost Function

The last thing that should be done to make the model (10) complete is to define the function (11). Therefore an analysis of the tracking process has to be done. This process consists in minimizing a distance between the robot and the target. Taking into account possible errors of prediction associated with “actions” of the second player a difference between the original target position (7) and the position resulted from one of the actions (13) should be involved into the cost function. Therefore the function has two-component form:

\[
I(a_1, a_2) = \hat{L}_{r,t}(a_1) + k_e \left[ \hat{L}_{r,t}(a_1) - \hat{L}_{\tilde{r},t}(a_1, a_2) \right]  
(15)
\]

where \( \hat{L}_{r, t}(a_1) \) is a predicted distance of the robot to the target determined from:

\[
\hat{L}_{r, t}(a_1) = \sqrt{\left( \hat{x}_r - \hat{x}_g \right)^2 + \left( \hat{y}_r - \hat{y}_g \right)^2}
\]

and

\[
\hat{x}_g = x_g + v(a_1) \Delta t \cos(\theta_g + \omega(a_1) \Delta t)
\]

\[
\hat{y}_g = y_g + v(a_1) \Delta t \sin(\theta_g + \omega(a_1) \Delta t)
\]

where \( v(a_1), \omega(a_1) \) denote values of linear and angular velocity set as a result of choosing the action \( a_1 \) from the action set \( A_1 \). Similarly \( \hat{L}_{\tilde{r}, t}(a_1, a_2) \) in (15) denotes a distance between the robot and the target if the first player applies the action \( a_1 \) and an error of prediction represented by \( a_2 \) appears:

\[
\hat{L}_{\tilde{r}, t}(a_1, a_2) = \sqrt{\left( \hat{x}_r - \hat{x}_g \right)^2 + \left( \hat{y}_r - \hat{y}_g \right)^2}
\]

and

\[
\hat{x}_g = \hat{x}_r + r(a_1) \cos(\alpha(a_1))
\]

\[
\hat{y}_g = \hat{y}_r + r(a_1) \sin(\alpha(a_1))
\]

The parameter \( k_e \) in (15) is used for tuning an influence of the possible errors of prediction to the control process.

Solution Method

The solution of the problem stated above and given by (10) is an action \( a_{1|0} \) that provides target-oriented motion of the robot. Moreover this action should soften results of inaccurate prediction. In case of high uncertainty good solution is min-max strategy that provides obtaining “the best from the worst” solution. Thus we have:

\[
a_{1|0} = \min_{a_1 \in A_1} \max_{a_2 \in A_2} I(a_{1|0}, a_2)
\]  
(17)

Applying (17) in each discrete moment of time \( t \in \langle t_n, t_{n+1} > \) the control vector (3) is obtained that provide reaching the target location (7) in the presence of uncertainty. Certainly (7) is only prediction of the target location calculated according the model (8) which often differs much from a real location. So if we want to reach the real location of the target: \( P(t_n) \) with a given accuracy a feedback mechanism has to be applied.

The Algorithm Overview

Overall process of tracking starts in the moment \( t_n=t_{0|0} \). Then, the model (8) of the target trajectory is determined using \( M \) past observations of the target state. Using this model an estimation of the future location (7) of the target after the time \( t=H \Delta t \) is determined. Next step is to plan a motion of the robot. The control vector (4) is determined that allow robot to reach the target in the time \( t=H \Delta t \). During execution of the robot motion according to (4) in each discrete moment of time \( t_n \) an accuracy of trajectory tracking (9) is checked. If it is lower then the acceptable value \( \varepsilon_T \) a correction of the model has to be made. A new prediction is made starting from the \( t_n=t_{k|0} \) and new control vector \( U \) of length \( H \Delta k \) is determined. The process is repeated until \( t_n=t_{M|0} \).

SIMULATION RESULTS

In order to verify the proposed approach great number of simulations were performed. All the experiments were implemented and executed with the use of MATLAB software. In this section results of a few selected experiments are presented that illustrate the work of the algorithm described in this paper. The process that we simulated was a typical tracking problem with additional time constraints.

The problem consists in tracking by the robot the target moving along unknown trajectory in this way to reach the target location after the given time \( t_{t|0} \). The result of the first simulation is presented in fig. 2. A plot of a part of the trajectory of the target \( P \) is depicted by a solid line. The robot is depicted by a polygon. The process of tracking starts in the moment \( t_n=t_{0|0} \). Initial state of the robot and the target are denoted by \( X(t_n=t_{0|0})=[40 50 0] \) and \( P(t_n=t_{0|0})=[62 97] \) respectively. Successive stages of prediction are depicted by crosses. The estimates of the target location for the time \( t_{t|0}=2[s] \) are depicted by circles and denoted by \( P_{pred1,2,3} \) respectively. The assumed accuracy of tracking \( \varepsilon_T \) is equal \( \varepsilon_T=4 \) [cm]. The resolution of discretization of the control space of the robot is set to \( \Delta \alpha=45[^\circ]/s \) for an angular velocity and \( \Delta v=5 \) [cm/s] for a linear velocity. In this experiment the algorithm needed 2 corrections to accomplish the task with the set accuracy in the set time \( t_{t|0} \). A plot of trajectory prediction errors is presented in fig 3. The second experiment (fig. 4) was performed under the same conditions as the first one. Also notation and symbols are the same as before. The target moves along a different trajectory. The horizon of prediction is longer and set to \( H=30 \) so the time...
It is worth noticing that the tracking process started in the moment after which the target changes its movement direction. So the prediction is not a simple task. As we can see the first estimate of the future location of the target is far from the real one. Then the first correction of prediction model is made after relatively short time. Next estimate is much better and robot follows the planned trajectory almost to the final position.

Figures 2: The Result of the First Tracking Experiment

Figures 3: A Time Plot of the Tracking Error $e$ Recorded During the First Experiment

CONCLUSION

In the paper the approach to a synthesis of the robust algorithm of tracking is presented. The discussed approach utilizes elements of the game theory. The model of the tracking process is built as a zero sum game between player (robot) and a nature (errors of prediction). The simulations that were run prove that presented approach leads to good results. Analyzing the results that we obtained a few conclusions can be drawn. The first thing that is easy to notice is that the tracking algorithm is convergent. In other words the control generated by the presented algorithm has the property that it still push the robot toward the target location. On the other hand the algorithm can not guarantee that the target will be reached in an assumed time. The main cause of this fact is the target trajectory is unpredictable. If the target changes a direction of its movement often and rapidly the prediction method does not give good estimation of the future location. Then the control of the robot can be out of feasible range. But great number of experiments proved that the algorithm worked well in most analyzed cases. Additional advantages of the proposed method of prediction as well as the tracking method are simplicity and robustness to uncertain data.

Figures 4: The Result of the Second Experiment

REFERENCES


ACKNOWLEDGEMENTS

This work has been supported by grant no 3T11A02628
APPLICATIONS OF THE HYPER-REDUNDANT ROBOTIC STRUCTURES TO HANDS OF THE PEOPLE WITH GREAT DISABILITIES

Viorel Stoian, Mircea Ivanescu, Elena Stoian, Cristina Pana
University of Craiova, ROMANIA
Mechatronics Department, Bld. Decebal, No. 107, 200440
E-mail: stoian@robotics.ucv.ro

KEYWORDS
Hyper-redundant structures, ER fluids, Hierarchical control, Fuzzy logic.

ABSTRACT

In this paper we present two applications of the hyper-redundant robotic structures to hands of the people with great disabilities. The first application refers to an electro-mechanical hand with unconventional actuators based on the properties of ER fluids to its fingers. The controller of the system is based on the viscosity control of the ER fluids by on electrical field. The second application refers to a medical glove with intelligent actuators for a hand with disabilities. The medical glove has got on outer superior face an intelligent actuator to every finger, which helps it to bend and to grasp different objects. On outer inferior face it has got a force distributed sensor system. The dynamic model of the outer superior face finger is determined and an approximate model is proposed. The two-level hierarchical control is considered.

MULTI-FINGERED HAND

ER Fluid Actuated Multi-fingered Hand-Model

An important disadvantage with traditional robot manipulators is that for a given gripper, only a small class of objects can be grasped. This limitation is sometimes overcome by equipping the robot arm with a tool changer, which allows different grippers to be used. Another disadvantage appears in assembly task, which require precise movements. For overcome these difficulties we propose to use multi-fingered robot hands as an alternative method. The extra degrees of mobility in a multi-fingered hand make it possible to grasp a large class of objects with a single end effector. This structure of the hand complicates both the kinematic and dynamic analyses of the system. In particular, since the hand is in contact with the object being manipulated, we must study the kinematics and dynamics of mechanical systems with contact constraints. Additionally, the increased degrees of mobility of the system increase the difficulty of planning a feasible grasp to perform a given task.

The most varied function of the hand is its ability to dynamically manipulate objects. Fingers motions may be repetitive and blunt (typing or scratching) or continuous and fluid with the rate and intensity of motion continuous controlled (writing or sewing). Prehension describes the ability of the fingers to grasp for holding, securing and picking up objects. There are many form of prehension: the grip, in which all fingers are used, the pinch, in which primarily the thumb and index fingers are used, the power grip, the precision grip, the power pinch, the precision pinch, hook grip and others. For hand prosthesis the pretension is the first goal. The physical structure proposed here can be a good solution.

Technological Structure of the Multi-fingered Hand Actuated with ER Fluids

[Diagram of a multi-fingered hand actuated with ER fluids]

Figure 1: Technological Structure of the Multi-fingered Hand

Figure 1 presents a multi-fingered hand technological model. The structure is a set of robots (fingers) which are attached to the end of a larger robot arm for the purpose of manipulation. Every robot-finger is a cylinder like that one above presented. The flexible pipelines are connected to hydraulic ON-OFF conventional valves.

Technological Structure of the ER Based Finger-Model

The general form of a finger model is shown in Figure 2. It is a cylinder made of fiber-reinforced rubber. There are three internal chambers in the cylinder, each of them containing the ER fluid with an individual control circuit. The deformation in each cylinder is controlled by an independently electro-hydraulic pressure control system combined with the distributed viscosity control of the ER fluid. In Figure 2a is presented the main segment structure.
The chambers of the segment have reinforced rubber walls with fibers on a circular direction. Thus, it is easy to deform it in the axial direction while it resists deformation in the radial direction. The cylinder can be bent in any direction by appropriately controlling the pressure in the three chambers. The electrical control of the ER fluid viscosity is obtained by an electrode network distributed on the length of the cylinder. The central electrode is connected to zero voltage and a high voltage controller supplies the peripheral electrodes. These electrodes can be individual controlled assuring a distributed voltage along the cylinder. A non-uniform viscosity is obtained (Figure 2b) along the s variable of the system.

Control System of the Fingers

The control system contains two subsystems: a conventional control loop of the pressure for the electro-hydraulic driving system and a variable structure control based on the viscosity control of the ER fluid (Figure 3a). The sliding mode in the control of the systems was proposed in (Schilling 1990) and developed for robot control in (Fu 1984). Unfortunately, the great number of variables increases the complexity of the problem and the control law remains in most cases only a theoretical solution. In this paper a new procedure based on a decomposition of the trajectory in a set of sequential switching lines and the control of the motion only by the viscosity control, directly on the switching lines, without to switch the control is proposed.

![Diagram](image)

The procedure is presented in Figure 3b. The motion is divided into two parts. The first part corresponds to the conventional control ($P_a$, $P_b$) and is obtained by the pressure control of the fluid (the control $T_1$). The second part is represented by the motion on the switching line $S_1$ by the control of the damping ratio. The procedure is repeated for each state plan. By this method the motion is decoupled into a number of lower dimensional problems. The switching manifold $S(e)$ is constituted by a sequential number of sub-manifolds $S_1(e)$, $S_2(e)$, ..., $S_N(e)$. A hierarchy of the control is imposed: $S_1 \rightarrow S_2 \rightarrow ... \rightarrow S_N$ meaning that the motion occurs on the switching surface $S_1 = 0$ only after it takes place on the manifold $S_2 = 0$, ..., $S_N = 0$. This control problem is reducible to a sequential analysis of the individual problems of a lower dimension. In each of these problems it is proved that the decoupled motion is determined only by the viscosity control (the damping ratio $\zeta$). The individual control in each of the arm segments is easily obtained by the structure proposed in Figure 2. The apparent viscosity of the fluid can be easily modified by the electric field.

Figure 3: Control System

Some Electro-Rheological (ER) fluids consist of a suspension of hydrophilic (water-retaining) particles suspended in a hydrophobic (water-repelling) dielectric fluid. This suspension is placed between electrodes for application of an electric field. In an electric field the particles polarize and inter particle forces then lead to the formation of chains, which tend to orient perpendicular to the electrode gap. With enough particles and a field of sufficient strength, the fibrils will bridge the gap and cause an appreciable increase in the viscosity of the suspension. There are a great number of papers (Brooks 1982, Duclos 1987, Duclos 1988, Hill and Van Steenkirk 1991, Ivanesco and Stoian 1995) which describe the two major methods of exploiting the ER effect in practical devices: "valve method" and "clutch method".
SPECIAL MEDICAL GLOVE

Physiological Aspects of Hand Functions

The hand functions as an effector organ of the upper extremity for: support, manipulation,prehension. As a support, the hand acts in a non-specific manner to brace or stabilise an object and, also, as a simple platform to transfer or accept forces.

![The Bones of the Hand](image1.png)

Figure 4: The Bones of the Hand

The ulnar nerve innervates the medial half of the flexor digitorum profundus muscle and is responsible for the sensation on the ulnar border of the hand and the ulnar one and one-half fingers. So, we propose the connection of the special glove with the median nerve and the ulnar nerve, because they realize the flexion motion of the hand in prehension. This is necessary, also, for maintaining the indispensable cortical representation of the motor and sensitive hand images. Generally the prehension is realized by thumb and the other four fingers. However the contribution in prehension is differentiated for each part of the hand. The fixed part is represented by the last carpal bones and the II and III metacarpal bones. The flexor and the extensor muscles of the carpometacarpal joint and the II metacarpal bone realize the stabilization of the fixed part in prehension. The most movable part is the thumb. It contributes with 50% in prehensive by his mobility and force. Index finger has an independent mobility, a high degree of sensitivity and has important contribution in the precise movements. In prehension the index indicates the direction of the grasp. The middle finger assures the force of the prehension and occupies the axis of the flexor muscles contraction. The ring finger is the assistant of the middle, as force and the second finger as sensitivity. It has a reduced mobility in prehension. The little finger increases the force in holding and securing the object of the prehension and, also, contributes to the stability of the grasp direction.

Physical Structure of the Glove

In Figure 5 is presented the physical structure of the special glove. On the superior faces of the glove fingers are fixed 5 tubes with have their structure presented in Figure 7 (hydraulic or pneumatic actuators) and on the inferior faces (at end of the glove fingers) are fixed strain-gauges for force measurement. The chambers of the segment have reinforced rubber walls with fibers on a circular direction. Thus, it is easy to deform it in the axial direction while it resists deformation in the radial direction. The cylinder can be bent in a plan (or in any direction, if it has 3 chambers) by appropriately controlling the pressure in the two (three) chambers (Figure 7). This tube has a hyper-redundant structure with a great number of points of mobility.

Hierarchical Control

The equations for the control of the arm parameters and for the control of the force offer a simple control for a Direct Sliding Mod Control (DSMC) (Schilling 1990). The DSMC is a control method which operates in two steps (Figure 9). First step assures the motion towards the switching line $S_q$ (or $S_p$). On the second step, when the trajectory penetrates the switching line, the damping coefficients are increased and the system is moving towards the origin, directly. The problem of controlling coordinating robotic systems with multiple chains in real time is complicated. A multiple chain hyper-redundant system is more complicated. A hyper-redundant robotic element is a physical system with a great flexibility, with
a distributed mass and torque that can take any arbitrary shape. Technologically, such systems can be obtained by using a cellular structure for each element of the arm. The control can be produced using an electro-hydraulic or pneumatic action that determines the contraction or dilatation of the peripheral cells. The first problem is the global coordination problem that involves coordination of several hyper-redundant elements in order to assure a desired trajectory of a load. The second problem is the local control problem, which involves the control of the individual elements of the fingers to achieve the desired position. The force distribution is a sub-problem in which the motion is completely specified and the internal forces/torques to effect this motion is to be determined. To resolve this large-scale control problem, a two-level hierarchical control scheme is used (Cheng and Orin 1991-a and b). The upper-level system collects all the necessary information and solves the inter-chain coordination problem, the force distribution problem. Then, the problem is decoupled into 5 lower-level systems, for every finger.

**Control System of the Special Glove**

The control problem asks for determining the manipulatable torques (control variable) $T^j_i$ such that the trajectory of the overall system (object and fingers) will correspond as closely as possible to the behavior. In order to obtain the control law for a prescribed motion, we shall use the inverse model. A closed-loop control system is used (Figure 8). A fuzzy control is proposed by using the control of the damping coefficient. The membership functions for control variables are shown in Figure 10 and for the output variables in Figure 11. According to the theoretical results obtained in the previous part of the paper, we can generate the control rules which establish a fuzzy control for a DSMC method (Table 1).
The main idea is to assure the normal control towards the switching line and direct control when the trajectory penetrates this line. A standard defuzzification procedure based on the centroid method is then used.

CONCLUSION

In this paper we present two applications of the hyper-redundant robotic structures to hands of the people with great disabilities. The first application refers to an electromechanical hand with unconventional actuators based on the properties of ER fluids to its fingers. The controller of the system is based on the viscosity control of the ER fluids by on electrical field. It is proved that the motion control can be divided into two parts: a conventional control determined by the second Lyapunov method for the pressure control of the ER fluid and a non-conventional control obtained by the viscosity control. It is proved that the stability of the motion on a switching manifold is assured if some constraints determined by the non-linear elements of the system are verified. Two control methods are proposed. The first is determined by a simultaneous control of the viscosity on the overall system. The second method is defined by a distributed control of the viscosity associated with a decomposition of the system in a finite number of lower dimension problems. These procedures are applied to a finger structure of the multi-fingered robot hands.

The second application refers to a medical glove with intelligent actuators for a hand with disabilities. The medical glove has got on outer superior face an intelligent actuator to every finger, which helps it to bend and to grasp different objects and on outer inferior face it has got a force distributed sensor system. The dynamic model of the outer superior face finger is determined and an approximate model is proposed. The two-level hierarchical control is considered. The upper level coordinator gathers all the necessary information to resolve the distribution force. Then, the lower-level local control problem is treated as an open-chain hyper-redundant structure control problem. The fuzzy rules are established and a fuzzy controller is proposed.

This paper presents some ideas of our team. Our future work will focus on the realization of new models, physical structures, testing and simulations.

REFERENCES

VISUAL INTERACTIVE SIMULATION
IMPROVING THE USE OF VISUAL INTERACTIVE SIMULATION AS A KNOWLEDGE ELICITATION TOOL

Professor Stewart Robinson
Mr. Ernie Lee
Warwick Business School
University of Warwick
Coventry, CV4 7AL
United Kingdom
Stewart.Robinson@warwick.ac.uk
Ernie.Lee02@wbs.ac.uk

Professor John S. Edwards
Aston Business School
Aston University
Birmingham, B4 7ET
United Kingdom
J.S.Edwards@aston.ac.uk

KEYWORDS
Visual Interactive Simulation, Knowledge Elicitation

ABSTRACT

Knowledge elicitation is a well-known bottleneck in the production of knowledge-based systems (KBS). This is mainly due to the tacit property of knowledge, which makes it difficult to be explicated, and hence analysed. Past research has shown that visual interactive simulation (VIS) could effectively be used to elicit episodic knowledge that is appropriate for machine learning (an AI capability that includes inductive learning) purposes, with a view of building a KBS. Nonetheless, the VIS-based elicitation process still has much room for improvement.

Based in the Ford Dagenham Engine Assembly Plant, a research project is being undertaken to investigate the individual/joint effects of visual display level and mode of problem case generation on the elicitation process. This paper looks at the methodology employed and some issues that have been encountered to date.

INTRODUCTION

Knowledge acquisition is a crucial stage in the development of a knowledge-based system (KBS). As a process, it involves eliciting the domain knowledge that a human decision-maker uses when solving a particular problem (knowledge elicitation), organising and structuring the elicited knowledge (knowledge representation), and then codifying the knowledge into an appropriate machine-executable format (knowledge execution). In spite of the technological progress in specialised tools designed to support the building of KBS, knowledge acquisition, especially knowledge elicitation, remains a well-known bottleneck. This is because the domain knowledge is usually qualitative in nature, which includes judgment, insight and informed opinions that are built on years of experience. As such knowledge is mainly tacit, the decision-makers may find it hard to verbalise. Consequently, their inability to explicate their thinking processes renders any attempt to elicit, and therefore acquire, the domain knowledge very difficult.

To date, various methods of eliciting tacit knowledge and expressing it into reader-friendly formats have been developed. These range from manual methods of interview and protocol analysis to machine-based methods like repertory grid and machine learning. Robinson et al. (2005) looked at linking Visual Interactive Simulation (VIS) and machine learning tools to improve unplanned maintenance operations in the Ford Bridgend Engine Assembly Plant. The authors showed that VIS could be used to elicit episodic knowledge in the form of decision cases that are appropriate for subsequent machine learning (a knowledge representation technique) purposes. Nonetheless, there are still issues that need to be explored, including how to exploit the functions of a commercial VIS package more fully to improve the quality and quantity of elicited decision cases. Specifically, the individual/joint effects of visual display level and problem case generation mode on the elicitation process need to be investigated.

This paper begins with the background to the research, including a discussion on the issues that help define the research aim and hypotheses. Then, the case study at Ford Dagenham Engine Assembly Plant which provides the research setting is briefly described. Following this, the methodology used to explore how VIS could be employed as an effective and efficient knowledge elicitation tool, with a view of building a KBS, is detailed. Finally, the paper concludes with an update of the project’s progress.

BACKGROUND

Robinson et al. (2005) carried out a collaborative study with Ford Motor Company (Ford) to investigate if VIS could be used to elicit decision cases from maintenance supervisors, such that they were good enough for learning the supervisors’ decision-making strategies. The aim of their study was to devise a VIS-based means for identifying and improving human decision-making. In their conclusions, the authors recognised that human decision-makers may take less realistic decisions in a simulated
environment, as they are quite likely to take greater risks when there are no real consequences to their decisions. In addition, the authors also recognised that human decision-makers may find it a very laborious and time-consuming experience to provide a full set of data, comprising of a very large number of useful decision cases. These issues provide the basis for the present research, where the aim is to investigate whether and how VIS could be used as an effective and efficient means to elicit decision cases.

In VIS-based knowledge elicitation, two levels of effectiveness can be conceived. First, the decisions made in the elicited decision cases are expected to bear close resemblance to those that a decision-maker will make in equivalent, but real situations (primary effectiveness). Second, the range of attributes and decisions collected in the elicited decision cases is expected to be wide enough to train meaningful decision-making models (secondary effectiveness). On the other hand, efficiency is the range of decision cases elicited in a span of time.

In the research, two specific factors are being investigated in terms of their influence on effectiveness and efficiency. The first is the level of visual display. The hypothesis is that a better visual display, with increased visual fidelity, leads to a greater level of primary effectiveness in knowledge elicitation (Hypothesis 1). In other words, the closer the visual display is to the real system, the closer the decisions will be to those taken in the real system. To test this hypothesis, the decision-makers will be presented with various 2-dimensional (2D), 3-dimensional (3D) and non-immersive virtual reality (VR) representations of the VIS model and asked to interact with the model. Here, a 3D representation consists of only 3-dimensional icons displayed against a plain background, with no perspective projection (where more distant objects are drawn smaller relative to those that are closer to the eye) or any other efforts at creating photo-realism. On the other hand, a non-immersive VR representation strives towards photo-realism, and the decision-maker is able to manipulate the view of the virtual environment by using a mouse.

The second factor is the mode of generating problem cases that are presented to the decision-makers. There are two modes: regular and non-regular. Under the regular mode, the model settings will be set such that it would simulate real-life operations as closely as possible. On the other hand, under the non-regular mode, the model settings will be set such that the probability of uncommon scenarios will be greater than in real life. The hypothesis is that the use of non-regular model settings to run the VIS model leads to a greater level of secondary effectiveness in knowledge elicitation (Hypothesis 2). In other words, non-regular model settings are expected to develop a larger variety of problem cases that would facilitate a wider range of decision cases to be elicited, and therefore a more meaningful decision-making model to be trained. To test this hypothesis, the decision-makers will be asked to interact with the VIS model running under the regular, and then the non-regular settings.

Finally, it is also hypothesised that the use of non-regular model settings will lead the decision-makers to produce a wider range of decision cases in a period of time (Hypothesis 3). In other words, non-regular settings will be more efficient for knowledge elicitation than regular settings.

**CASE STUDY: FORD Dagenham Engine Assembly**

![Figure 1: A Schematic of the Hot-test Area](image)

Set in an engine assembly plant in Dagenham (East London), the research is based on the hot-test operations of the ‘Puma’ diesel engine assembly line. A schematic of the hot-test area is shown in Figure 1. Engines arrive from the main assembly line (Assy line ‘B’) where they are loaded onto a hot-test ’platen’ (a metal pallet). They are then allocated to one of the 20 test cells, where they are riged to a testing machine and run for a few minutes. Following the test, failed engines are sent to a repair loop for rectification, while engines that passed the test are sent to a finishing area (ATD). The key decision-maker here is the ‘switch operator’, who is responsible for the allocation of engines to hot-test cells. His objective is to maintain a smooth workflow with no bottlenecks, and ensure that the workload is equitably distributed among all operational hot-test cells.

**RESEARCH METHODOLOGY**

The research methodology (to be) employed can be broken down into seven main stages: understanding the decision-making process; building a ‘game’ model; pilot testing the ‘game’ model; eliciting decision cases; learning the decision-makers’ decision-making strategies; validating the strategies; and analysing the elicitation, learning and validation data. These stages are described further below.
Stage 1: Understanding The Decision-making Process (Completed)

The purpose of this stage is to lay the groundwork for the research. To begin, there is a need to gain a high-level understanding of the manufacturing system (i.e. hot-test area in the Dagenham engine plant) and the circumstances in which a decision-maker (i.e. group leader, or switch operator) is required to make decisions. In addition, the component parts (decision variables, decision options, attribute variables, and attribute levels) of the decision-making process are identified. In effect, this is akin to the ‘broad and shallow’ phase in Barrett and Edwards (1995), where the priority is to extend the breadth of background information as wide as is feasible. As no technique alone is sufficient to elicit all types of information (Coffey and Hoffman 2003, Rugg et al. 2000, 2002), several techniques that complement each other in gaining a broad overview in this stage are used. The techniques used here include document analysis, unstructured and semi-structured interview, and observation interview.

First and foremost, paper documents like plant layout and versatility charts (detailing the hot-test activities that each personnel is qualified to undertake) are reviewed to obtain a quick introduction to the hot-test area. The plant layout serves to help put the hot-test area in the context of the entire engine plant. It informs on the processes before an engine enters and after the engine leaves the hot-test area. In addition, it also provides a record of the physical entities that make up the engine plant, specifically the hot-test area. Versatility charts serve to provide information on the manpower status of the hot-test area. They inform on the type of responsibilities that hot-test personnel in every shift can undertake. This information is especially useful when the planning of data collection commences.

Next, informal interviews are conducted with the group leader and switch operators to collect any other relevant undocumented information, and then to clarify any queries that are formed after reviewing the documents and interview materials. In addition, such preliminary interviews also help to create opportunities for establishing good rapport with the decision-makers, whose co-operation in subsequent data collection exercises is crucial. Initially, unstructured interviews are used where the decision-makers are given the freedom to cover topics that they deem fit. This is because at this early stage, a researcher who is not familiar with the hot-test area will not have enough background information to ask specific questions (like in structured interviews), or even work to cover a list of topics (like in semi-structured interviews) in an interview session. After a few unstructured interview sessions, the researcher should have collected sufficient information from both documents and interview materials to form a clearer picture of the hot-test operations. At this point, a few queries might have surfaced, and these may be answered via another few rounds of semi-structured interviews.

Last but not least, observation interviews are conducted as a ‘catch-all’ attempt to collect additional information that is neither documented nor communicated in the earlier efforts. Here, the researcher will observe and record the decision-makers’ activities in the hot-test area. If the researcher has any queries in respect of his/her observations, s/he will clarify them with the decision-makers at the first instance. Such queries will range from the reasons behind the observed activities to the consequences as a result of them. In this way, the researcher may have conceived a rough idea of the decision-makers’ decision-making strategies. In addition, questions regarding the physical and logical design of the hot-test area may also be asked. The latter information is especially important during the stage of building a ‘game’ model of the hot-test area. The aim of this ‘observe-query-observe’ activity sequence is to verify any assumptions that are made during the document analysis and interview efforts, and to reinforce the researcher’s understanding.

Stage 2: Building A VIS ‘Game’ Model (Completed)

Before plunging into building a model of the hot-test operations, the nature of decision-making in the system needs to be scrutinised first, in order to ensure that the model subsequently built is fit for the research purpose. Using Mintzberg’s (1973) classification of management roles, the decision-makers’ main role is determined as that of a resource allocator. In what is essentially a passive system, the decision-makers either make proactive decisions to pre-empt a bottleneck from emerging, or passive decisions to maintain status quo (decision of ‘no decision’). Hence, there is a need to build the model in such a way as to capture both types of decisions.

The concept of a VIS ‘game’ model fulfils this need. The model is a ‘game’ because like any simulation game commercially available in the market, it is expected to run passively according to the default settings until the user proactively changes the settings. To capture the pro-active decisions, the decision-makers will be shown a running ‘game’ model, and they will be allowed to intervene and stop the running ‘game’ momentarily to amend the allocation decisions as and when they deem fit. When an amendment is made, the set of new decision values and corresponding attribute values will be saved as a decision case, which will be required later for learning the decision-making strategies. To capture the passive decisions, the ‘game’ model could be set to perform background saves of current decision values and corresponding attribute values at random intervals of simulated time.

Using the information collected in Stage 1, three representations (2D, 3D and VR) of the ‘game’ model are built. In this case, as Ford already has a current and detailed model of its Dagenham engine plant developed in Witness (a VIS software), there is no need to build a new ‘game’ model from scratch. Instead, Ford’s existing model will be adopted and adapted for the purpose of the research.
Since Ford’s model is still currently in use, it can be assumed that the model is current. In adapting the Ford model, extra care should be taken to ensure the ‘game’ model’s fidelity is on a what-you-see-is-what-you-get basis. That is, the ‘game’ model should provide neither more nor less information than the decision-maker is able to obtain/perceive in the real working environment. Also, in creating the different representations, great care should be taken to ensure that the availability of information in all three representations is consistent.

**Improving The VIS ‘Game’ Model’s Utility**

To make the game as realistic as possible and hence remove any factors that might unnecessarily influence the fidelity of the decision-making process, one would expect the model to run at the same speed as the real system. However, problems might arise if real operations take a long time to complete, or if they are to be simulated for a long period of time (say, an eight-hour shift). As such, there is essentially a conflict between keeping the game faithful to the real system and maintaining the experiment’s expediency. Since access to the switch operators is limited, game faithfulness is traded off for higher experiment expediency by running the model at a pace that is much faster than real time.

As Ford’s VIS model spans the entire engine plant, it is expected to run relatively slowly. This, in turn, is expected to slow down the entire elicitation process. Hence, every effort should be made to improve the VIS model’s utility (run-speed). A possible solution is to simplify the VIS model as much as possible without compromising the model’s validity or credibility. An option for simplifying the model lies in splitting the original model into two parts, comprising a sub-model of pre-hot-test operations (Model A), and another sub-model of hot-test operations and post-hot-test operations (Model B). In this case, as Model A runs, its output data are collected and written to a data file. The contents of the data file are then used as input data when Model B is run. Nonetheless, there is a limitation to just using the raw contents of the data file as input data for Model B. As the input data to Model B is going to be an experimental factor, it may be more advisable to use the raw contents to construct a pseudo-empirical distribution of inter-arrival times that can be manipulated easily for subsequent experimentation. The distribution thus constructed is not strictly considered a genuine empirical distribution, as the data file contents itself are output data from another simulation (Model A, in this case) as opposed to historical data collected manually from the engine plant. For the research’s purpose, approximately 33,000 inter-arrival times have been collected from ten simulation runs of Model A. Each run simulated operations for a full week.

**Improving The VIS ‘Game’ Model’s Logical Design**

Next, the remaining model will be scrutinised for its logical design. This is because the original model is built for a purpose different from the research. In the original model, there may be some operations that require human supervision and intervention in practice (such as by the switch operator), but which have been automated by pre-set rules defined by the model builder. Hence, there is a need to remove these pre-set rules that automated the switch operations, and re-establish the facility for the switch operator’s intervention/proactive decision-making efforts. Also, for data collection purposes, the ‘game’ model needs to be programmed in such a way as to be capable of recording the decision values and corresponding status of the model (attribute values) at each intervention.

**Stage 3: Pilot Testing The VIS ‘Game’ Model (Completed)**

Prior to rolling out the adapted ‘game’ model to collect data, it should be pilot tested. The approach to pilot testing may be adapted from those applied to questionnaires (see for instance, Saunders et al. 2003). The purpose of the pilot test is to refine the ‘game’ model so that the decision-makers will have no problems in using the ‘game’ model and there will be no problems in recording the data. In addition, it will enable the researcher to obtain some assessment of the ‘game’ model’s validity and credibility.

**Stage 4: Eliciting Decision Cases (Featuring Hypothesis 1 – Partially Completed)**

After pilot testing, the ‘game’ model will be formally employed to collect data under the different research experimental settings (i.e. factor-level combinations). As mentioned earlier, the research will be exploring if and how factors like visual display level (set at levels: 2D, 3D, and VR), and problem cases generation mode (set at levels: regular model settings, and non-regular model settings) will influence the effectiveness and efficiency of VIS-based knowledge elicitation. Hence, there are altogether six factor-level combinations to investigate. From Stage 1, it is known that there are eight decision-makers who are qualified to perform the switch operations and available to help. They will be used carefully in collecting data.

In this stage, Hypothesis 1 will be tested. Under Hypothesis 1, the view is that increased visual fidelity in VIS-based knowledge elicitation will improve the KBS’ fidelity to real-life decision-making (primary effectiveness). This hypothesis can be tested using a single blind evaluation test (Jones and Miles 1998), where a selection of decision cases collected earlier from every decision-maker under each factor-level combination will be presented back to them in a randomised order. Every decision-maker will then be asked to rate how strongly he agrees with each decision case.

**Stage 5: Learning The Decision-makers’ Strategies (Featuring Hypothesis 3)**

This stage sets out to learn resource allocation strategies from the decision cases collected earlier. They may be
used either as direct references to support similar decisions or to induce rules (Turban et al. 2005). The former, called case-based reasoning (CBR), adapts solutions used to solve old problems for use in solving new ones. The latter, called inductive learning or rule-based reasoning (RB), allows the computer to examine decision cases and generate rules that can be inferred to solve problems. Though it is not an objective to compare the quality of learning managed by CBR and RB systems, it may be interesting to find out if decision cases provided by VIS-based elicitation are more suitable for CBR or RB systems.

To support the testing of Hypothesis 3, RB will be used to learn the resource allocation strategies. Under Hypothesis 3, the view is that non-regular model settings will lead to a wider range of decision cases being produced in a period of time. Intuitively, the latter is expected to have a positive correlation with the number of rules generated. As such, by analysing the number of rules generated for each decision-maker under various factor-level combinations and the corresponding times taken to elicit the decision cases, some light might be shed on the hypothesis.

Stage 6: Validating The Decision-makers’ Strategies (Featuring Hypothesis 2)

The resource allocation strategies learnt in the preceding stage will be primarily examined using the principle employed in the Turing Test. The Turing Test is a test designed to determine whether a system exhibits intelligent behaviour (Turban et al. 2005). According to this test, a system can be considered smart only when an individual cannot identify the system, while conversing with both an unseen human being and an unseen system. As applied to the research, the relationship between decisions developed by the system and decisions developed by decision-makers will be measured. In essence, the system is treated like a black-box, where only the attributes and decisions are analysed, and the system processes are not investigated. As such, a smart system will arrive at the same decisions as the decision-makers, when both are provided with similar sets of attributes.

Hypothesis 2, where the view is that using non-regular model settings in VIS-based knowledge elicitation will improve the KBS’ ability to make meaningful decisions in more varied scenarios (secondary effectiveness), can be tested through the assessment for intelligent behaviour. Logically, a smart system is contingent on its knowledge base being developed from data that is both accurate and covers a wide spectrum. As accuracy of data has been dealt with in Hypothesis 1, and on the premise that the decision cases are indeed of high fidelity (i.e. Hypothesis 1), then simply proving that the system is smart would imply that the decision cases do cover a range that is sufficiently wide to learn the decision-making strategy.

Stage 7: Analysing The Elicitation (Stage 4), Learning (Stage 5) And Validation (Stage 6) Data

Eight sets of data will be collected from each factor-level combination in the elicitation, learning and validation stages. As the sample size is too small, appropriate small sample statistical tests and non-parametric tests are most likely to be used.

PROGRESS UPDATE

To date, Stage 1 to 3 (understanding the decision-making process, building the ‘game’ model, and pilot testing the ‘game’ model) are completed and the research is on its way to completing Stage 4 (eliciting decision cases). Investigations in Stage 1 have uncovered the following decision variables and attributes:

**Decision variables**

i. Pre-set allocation to straight, left or automatic
ii. Switch test cell on/off
iii. Manual override to eject engines from waiting stands
iv. Allocate a specific engine to a specific test cell

**Attributes**

i. Engine type to be tested
ii. Engine type currently being tested in a cell
iii. Operational status of a test cell
iv. The number of engines on each section of conveyor
v. The number of engines serviced by a cell operator in a period

In addition, the detailed VIS model provided by Ford has been adopted and adapted into an interactive ‘game’. As illustration, the 2D, 3D and VR ‘game’ models are shown in Figure 2, Figure 3 and Figure 4 respectively. In a typical elicitation session, the current attribute values would be displayed/represented in the model, and would be constantly updated while the model is running. As and when the attribute values warrant a change in the current decision values, the decision-makers would be able to control the model and do so by using the ‘game’ toolbar below the model. For instance, in Figure 2, the decision values of the circled model parts could be controlled by the toolbar icons. Whenever the current set of decision values is changed, the new set of decision values and corresponding attribute values will be saved as a decision case for subsequent machine learning.

Several trips have been made to the Dagenham engine plant, including one where the prototype ‘game’ was demonstrated to two decision-makers. To establish its content validity, the decision-makers were allowed to play with the prototype ‘game’ for a while, before their comments/suggestions were sought. Their feedback to the ‘game’ has so far been positive. Following this, even more trips have been made to the engine plant as part of the planned data collection process. At the time of writing, the elicitation sessions are almost completed.
methodology are completed and the fourth stage is near completion. Subsequent work will include eliciting a complete set of decision cases (data) from each decision-maker, and analysing them.

Although the research is based on a specific case study, it is expected that certain generic lessons will be learned that would be useful to modellers in other contexts. For instance, the likely effects of visual display on decision-making quality (fidelity), and that of the range of scenarios generated on data quality (extensiveness).

ACKNOWLEDGEMENTS

This work is being funded by the UK Engineering and Physical Sciences Research Council (EPSRC) through the University of Warwick Innovative Manufacturing Research Centre. The authors also acknowledge the support of Ford Europe and the Lanner Group with respect to this research.


REFERENCES


AUTHOR BIOGRAPHIES

STEWART ROBINSON  is Professor of Operational Research at Warwick Business School. He holds a BSc and PhD in Management Science from Lancaster
University. Previously employed in simulation consultancy, he supported the use of simulation in companies throughout Europe and the rest of the world. He is author/co-author of three books on simulation. His research focuses on the practice of simulation model development and use. Key areas of interest are conceptual modelling, model validation and output analysis. Stewart is also carrying out research into understanding and modelling human interaction with systems. Home page <www.bitinternet.com/~stewart.robinson/ST_HTM>

**ERNIE LEE** received a BAcc from the Nanyang Technological University, Singapore. He has also completed an MSc in Operational Research from the Lancaster University Management School. At present, he is studying for his PhD at the Warwick Business School.

**JOHN S. EDWARDS** is Professor of Operational Research and Systems at the Aston Business School in the UK. He holds a BA and MA in Mathematics, as well as a PhD in Operational Research (Manpower Planning) from the University of Cambridge. His interest lies in understanding how people use models and systems to help them do things. For many years, he has studied the development and uses of knowledge based systems and decision support systems in management and administration, especially the integration of knowledge based systems, decision support systems and conventional information systems.
Position-aware Service Provision with RFID

Alessandro Genco, Salvatore Sorce, Rossella Battaglia, Eleonora La Spina
Dipartimento di Ingegneria Informatica – DINFO
Università degli Studi di Palermo
Viale delle Scienze, edificio 6
90128 Palermo - ITALY
{genco, sorce, battaglia, laspina}@studing.unipa.it

KEYWORDS
Augmented reality, Mixed reality, Radio Frequency Identification, Ubiquitous Computing, Pervasive Systems, Context-aware Services, Smart environments

ABSTRACT

Radio Frequency Identification (RFID) has recently received a lot of attention as an augmentation technology in the pervasive computing domain. In this paper we present a brief introduction about RFID technology and its possible contribution for the future achievement of a system that is pervasively and unobtrusively embedded in the environment, completely connected, intuitive, effortlessly portable, and constantly available.

The goal of this paper is to illustrate the potentialities of RFID when this technology is used to implement augmented reality (AR) and mixed reality (MR) systems. To this end, we present the results of our conceptual design and the implementation of a pervasive application using smart identification technologies.

INTRODUCTION

RFID technology represents a suitable means to bridge the physical and virtual world in an invisible or at least unobtrusive manner (Want et al. 1999). Like many other technologies, low-cost Radio Frequency Identification (RFID) systems will become pervasive in our daily lives when affixed to everyday consumer items as “smart labels”. Embedding computation, in fact, would enable people to move around and interact with computers more naturally than they currently do. One of the goals of ubiquitous computing is to enable devices to sense changes in their environment and to automatically adapt and act accordingly to these changes based on user needs and preferences. Some simple examples of this type of behavior include GPS-equipped automobiles that give interactive driving directions, and RFID store checkout systems.

AR and MR aim to make everyone feeling comfortable with their daily surrounding reality: natural objects are perceived as we are usual to do, even if they belong to an artificial system, which mixes real physical beings with virtual logical ones (Want et al. 2002). According to this vision, we deal with object tagging, which is an enabling concept for many interesting pervasive computing applications. By attaching small electronic tags to physical objects, these can be automatically identified and located when brought into the vicinity of a tag detection system.

In this paper we present the design and implementation of a system for location-related information provision.

RFID BACKGROUND AND FEATURES

In the Ubiquitous Computing vision, technology is seamlessly integrated into the environment and provides useful services to humans in their everyday lives (Weiser 1991). Radio Frequency Identification (RFID) technology can significantly contribute to the realization of such a vision, as demonstrated by many researchers over the years. Some examples are the Magic Medicine Cabinet (Wan 1999), the augmentation of desktop items (Want et al. 1999) and smart shelves (Decker et al. 2003). These prototypes show that RFID technology has many benefits over other identification technologies because it does not require line-of-sight alignment, multiple tags can be identified almost simultaneously, and the tags do not destroy the integrity or aesthetics of the original object. Due to the low cost of passive tags and their power-less operation, there are also some weaknesses associated with RFID-based object identification, as shown in (Floerkemeier and Lampe 2004).

A Radio Frequency Identification (RFID) tag is an electronic device that holds data. Typically these tags are attached to an item and contain a serial number or other data associated with that item.

When an RFID tag passes through the electro-magnetic zone, it detects the reader's activation signal. The reader decodes the data written in the tag’s microchip and the data is passed to the host computer for processing (Fig. 1).

Fig. 1: Basic principle of a RFID system
RFID tags are categorized as either active or passive. Active RFID tags are powered by an internal battery and are typically read/write, i.e., tag data can be rewritten and/or modified. An active tag's memory size varies according to application requirements. Passive RFID tags operate without a separate external power source and obtain operating power generated from the reader. Passive tags are consequently much lighter than active tags, less expensive, and offer a virtually unlimited operational lifetime but they have shorter read ranges than active tags and require a higher-powered reader.

Developments in RFID are yielding larger memory capacities, wider reading ranges and quicker processing, making it one of the fastest growing sectors of the radio technology industry (Finkenzeller 2004).

RELATED WORKS

RFID applications in use today exploit the full range of tag technology, from cheaper tags to highly expensive miniature sensor/transponders. Animals and livestock have been tracked using RFID technology for decades, but RFID has recently become a technology of choice for tracking humans. This is realized by using “smart cards” with the size of credit cards, active RFID tag bracelets, and even tiny chips embedded in the skin.

Europe's largest amusement park, Legoland in Denmark, uses active RFID tags contained in bracelets and Wi-Fi networks to help parents track their children through the park. (Laurie Sullivan 2004).

The PRISM system, developed by Alanco Technologies, Inc. for use in correctional facilities, uses a tamper proof RFID-enabled wrist bracelet to monitor the location of prison inmates in real time, reducing instances of prison vandalism and other unruly behavior (Alanco).

The United States Transportation Security Administration (TSA) is considering the use of RFID-tagged airline boarding passes (Brewin 2004).

Applications that are not initially designed to track individuals, such as the RFID-based electronic highway toll collection system EZ Pass, might nonetheless make human tracking possible (Gibson 2003).

RFID manufacturer Applied Digital Solutions (ADSX) has developed a passive chip the size of a pen point which is implanted in the human body. The VeriChip Personal Identification System is designed for use in a variety of applications including financial and transportation security, residential and commercial building access, military and government security. A nightclub in Spain began using the VeriChip system in March 2004, to improve access for VIPs and allow them to pay for drinks without cash or credit cards (Purohit 2004).

SYSTEM OVERVIEW

We propose a system whose aim is to provide people with information related to their position within an environment, using a PDA equipped with a RFID compact flash reader module. This schema is similar to the one foreseen by Ohashi (Ohashi 2005).

Our system has a client/server architecture. A central computer runs the server application, which can simultaneously keep multiple connections with PDAs (Fig. 2). The PDA runs the client application, which communicates with the server using the TCP/IP protocol (over a Wi-Fi connection) and asynchronous sockets. This way the system becomes easily scalable.

The first step to achieve the system goal is to tag any object of interest (or either “point” of interest) within a given environment. People walk across such environment with a PDA running our client application.

When an RFID tag is detected by the reader module on the PDA, data read from the tag is passed to the PDA to be processed. Then the client application running on the PDA sends the detected tag ID to the server via the wireless connection.

The server application uses the received ID to query a locally available database, in order to obtain information about the tagged object. Once information have been retrieved, the server application send them back to the client application on the PDA which shows them on its display.

![Fig. 2: System architecture overview](image)

The link between the information sent to the PDA and the user's position is given by the detected ID.

In our experiments we used commercially available RFID readers and tags. We exploit the limited range of RFID passive tags read by a compact flash reader (50 cm. or less) to define areas in which users can be detected. This way we are sure that whenever a PDA detects a tag, the tag is actually in the PDA's neighborhood and we send information that are actually related to the PDA's (and to the PDA's owner too) position.

In other words, the limited reading range of passive RFID tags limits the maximum position estimation error. This result can be taken into account because we are only interested in position detection within a number of spot-areas, each defined by the RFID range, and each corresponding to an object of interest within the environment.

Actually, the PDA also runs a management application which allows to send some commands to the reader, such as setting instructions. These script commands, which are sent
directly to the reader device through COM port, allow us to set the operating instructions with the possibility, i.e., to exclude or include a particular tag type from reading.
We decided to implement a client/server architecture with a centralized database, because the tag can only store a limited amount of data and therefore information can be rather limited to the most basic aspects of the object to which the tag is attached. We use the tag’s ID as a primary key within the database, so we can store (and, of course, retrieve) a large amount of additional information about the tagged objects. This way every object within a given environment can be “augmented” with additional semantics.
We implemented all the software parts of the system described above, namely the database, the server application, the client application which runs on the PDA. For the client side, we also implemented a dll which is used to interact with our CF-RFID reader from any user-written application. The dll allows programmers to implement any kind of interaction with RFID tags through the reader, thus permitting to change the system’s application field.

CASE STUDY

The proposed system can be used in several application environments, such as museums (information about artifacts and works of art, Fig. 3), factories (information about a tool, such as operating instructions or work-to-do), schools and universities (information about classroom schedules, teachers’ hours).

Fig. 3: Example of interaction between an user and a tagged object (in this case a painting in a museum)

In particular we carried out some experiments within our Department, where a service-provision framework based on personal mobile devices is running (Genco et al. 2005), in order to test our design choices, given that it is very straightforward to adjust the application environment.
Such a framework is designed to supply students with ad-hoc information and high-level services by means of their own personal mobile devices.
In our test environment, a possible list of services users could be supplied with, is:
• documents request;
• examination enrollment;
• access to teaching information, such as courses time or exam dates;
• request for teaching aids, to be directly downloaded with the mobile device or to be sent to an e-mail address;
• access to location- and time-related information, such as department directories, professors receiving hours, classroom schedules.

It has to be noticed that our system is not intended to continuously track people position within a site, but only to sense their position at given spot-areas near selected points of interest. Each spot-area is defined by the RFID tag-reader range.
We used WiFi-enabled Toshiba e740 and Asus A730W PDAs equipped with CF-RFID readers (Fig. 5). When a RFID tag is detected, the application running on the PDA queries the database via the wireless network.
With this kind of hardware, we set up a test environment with two Points of Interest (POIs). In more detail, the two POIs are placed along a corridor, at two different positions (corresponding to two classrooms) on the same side (Fig. 4). We use information about students position in order to supply them with current class information, classroom occupation schedule, and links for educational materials as described in (Genco et al. 2005). Information about students position detected with RFID readers is used to select suitable data and to trigger its transmission.

Fig. 4: Test environment with two points of interest on the same side of a corridor

We saw that students were feeling this way of service provision very useful and more intuitive than a common web interface. Furthermore, it has to be noticed that some kind of information can only be provided via our system
due to its own nature. This is the case of real-time changing information, such as the current course in a classroom or schedule variations due to professors engagements. Of course, visitors of a museum or workers of a factory could take more advantage than students from our proposed system. In fact, they could obtain a more large and helpful variety of data related to their current position.

CONCLUSIONS

The experiments carried out demonstrated the effectiveness of our design choices, and confirmed that RFID can be successfully used as a cheap and reliable technology for proximity-based positioning detection, to be used as base for context-aware service provision systems.

We are currently working on a better tuning of database queries, by considering the detected ID and the user profile too. This way the system could provide users with more suitable information, which can be related not only to the user’s position, but also to his personal profile (such as skills, education, preferences, language) stored on his own personal device.

REFERENCES

AIM, Association for Automatic Identification and Mobility, www.aimglobal.org
Purohi C., “Technology Gets under Clubbers’ Skin”, CNN, June 9, 2004
Want R., Pering T., Borriello G., Farkas K., “Disappearing Hardware”, on IEEE Pervasive Computing, Volume 1, Issue 1, Jan.-March 2002 Page(s): 36 - 47

BIOGRAPHY

ALESSANDRO GENCO is an associate professor at the Department of Information Engineering and the coordinator of the PhD School of Computer Engineering at the University of Palermo. His research interests include mobile networks and pervasive system middleware design for augmented reality. He received his degree in mathematics from the University of Palermo.

SALVATORE SORCE is an instructor and research collaborator at the University of Palermo. His research interests are in mobile agents in collaborative augmented reality environments and in parallel and distributed applications, pervasive computing, wireless networks and communications, wearable computers, positioning systems, and personal mobile devices programming for pervasive systems. He received his MS and Ph.D. in computer engineering from the University of Palermo.

ROSSELLA BATTAGLIA and ELEONORA LA SPINA received their degree in computer engineering from the University of Palermo. They are currently involved in national research project in mobile devices programming for remote service access.
SIMULATION OF ECOLOGICAL SYSTEMS
Optimizing ecology-friendly drawing of plans of buildings by means of grammatical evolution

Arturo de Salabert, Alfonso Ortega, Manuel Alfonseca

Escuela Politécnica Superior – Universidad Autónoma de Madrid

ABSTRACT
We explore the application of grammatical evolution to the automatic generation of plans of building with constraints. A BNF is presented that guarantees the conversion of the genetic code into a well formed geometrical figure or phenotype. The validity of the approach is demonstrated, its limitations are analyzed and new evolutionary techniques are suggested for future work in this area.

1. INTRODUCTION
In the last decades, computer science has witnessed an interesting search for inspiration in Biology to help solve many different sorts of problems. Evolutionary computing, one of the results of this search, has created a number of constructs which have been applied successfully to multiple problems. Genetic programming (GP) has proved to be an effective and efficient generic search and optimization method in a wide variety of situations, while grammar evolution (GE), introduced by [Oneill 2001], has been elegantly applied to automatic programming [Oneill 2003] or to automatic music composition [Ortega 2002].

When applicable, GE has several advantages with respect to GP. From our perspective, the most relevant is the separation of the genetic code from the phenotype. This provides a great degree of freedom in developing or adapting the phenotype, without modifying the low level genetic processes. This is another example of the “onion principle”, whose advantages are well known in fields like communications (ISO layers) or software engineering (OS levels).

Grammar-based drawing has been proposed [Ortega 2003] for the drawing of fractals by using a turtle like coding and L-grammars, but these studies did not need to address the two limiting problems which are encountered in our case: closure and the search space, as will be explained later.

Our approach starts by considering the plan of a building, not just as a group of lines, but as the result of a process of adapting to an environment certain user requirements, which impose a number of physical limitations. From the biological perspective, developing a building would be similar to growing a tree or a plant from a seed. The genetic code carries the characteristics of the specific variety of tree (the requirements for the building) while the place where the seed has fallen will set the limitations or constrains to its growth. The final objective of such an approach would be to automate the generation of plan drawings adapted to the environment, while, at the same time, fulfilling the requirements imposed by the designers. As a result of this process, we would have a variety of seeds that would compete between them to develop the best solution. Buildings of different shapes and characteristics would grow from these seeds, adapted to the particular circumstances, and the designer would have a set of choices to start the final draft, or even a set of solutions to choose from.

In this exercise we have chosen plans of buildings because it is at the same time a complex problem, rich in features and full of constraints, and also a familiar one, easy to explain. Machine drawing, for instance, would be another comparable problem, but not as intuitive.

2. GRAMMAR EVOLUTION
Grammatical Evolution [Oneill 2001] is a grammar based, linear genome system, which has been applied in the area of Automatic Programming to automatically generate programs or expressions in a given language that solve a particular problem. In Grammatical Evolution, the Backus Naur Form (BNF) of the grammar of the language is used to describe the output produced. Different BNF grammars can be used to automatically produce code in any language.

In Grammatical Evolution, the genotype usually is a string of 8 bit binary numbers generated at random, treated as integer values from 0 to 255. The phenotype may be a running computer program generated by a genotype-phenotype deterministic mapping process which translates the genotype, codon by codon from left to right, in the following way: the mapping begins with the axiom; at each step, the leftmost non terminal symbol of the current sentential form is chosen; the current codon is used to select the rule (among those available for that symbol) which is numbered with the current codon mod the number of rules for the current non terminal; a new sentential form is ob-
tained by applying the rule to the non terminal. When the string of integers in the genotype is exhausted before the phenotype has been completely generated, a biologically inspired wrapping mechanism is used to reuse the integers, similar to the gene-overlapping phenomenon observed in many organisms in nature. The mapping benefits from genetic code degeneracy, i.e. different integers in the genotype generate the same phenotype, according to Kimura’s neutral theory [Kimura 1983].

In Grammatical Evolution, standard genetic algorithms are applied to the different genotypes in a population, using the typical crossover and mutation operators. For each domain, one must design the proper fitness function, which will be used by the genetic algorithm to perform selection.

3. PROBLEM DESCRIPTION

We suggest a think big, start small approach, and will start by the low hanging fruits, focusing first on a simple problem: how to draw the largest geometrical figure in a plot of land covered with trees. We will assume that local regulations penalize felling and limit the maximum size, while the minimum sizes are determined by practicality. Further constraints are easy to add: cost, orientation, shape... but our initial objective is not to make a realistic plan, but to experiment with the evolutionary approaches.

As our first simple case we’ll assume a square plot of land of side n with some trees inside of side 1. The first objective of our evolutionary algorithm is to draw the largest figure with the minimum felling.

Of course, we could find very efficient geometrical solutions to this problem, but keep in mind that our objective is more ambitious. Our toy system will be implemented subject to the following considerations: A flexible case should not have a predefined shape; therefore the genetic code cannot have a limited length. There are several conditions to be fulfilled (which could change) and an obvious closure problem. Leaving all this to the fitness function is a significant overload. The phenotype correctness should be insured without affecting the genetic process. Complexity should be scalable without changing the basic genetic rules. Enable evolution and adaptation to the environment.

The GE system proposed here provides an answer to each of the previous points: Variable length genetic strings can be used, decoupled from the actual drawing. The suggested Backus-Naur (BNF) grammar takes care of the fulfillment of any conditions as well as the closure problem. This is done without putting any extra load on the fitness process. The unconstrained search of genotypes is compatible with the production of a syntactically correct plan to be evaluated, as the genetic process (selection, crossing-over, mutation) does not operate on the actual plan, but on the strings that make up the genotype. The procedure can be used to generate any type of drawing. Evolution and adaptability to the environment are enabled at the phenotype building time, by means of a two level process: coding the phenotype and expressing it within the constraints imposed by the environment.

In addition, genetic diversity and resilience to mutation are obtained by degeneration, and last but not least, it will be shown that a significant reduction of the search space is obtained.

4. SOLUTION DESCRIPTION

The proposed GE approach consists of the following components:

- A genetic algorithm based on the random creation of a first generation of individuals, which gives rise to their evolution based on selection, coupling and reproduction.
- A BNF grammar representing the language that will be used to generate a family of plans.
- The phenotype developing process, which has been divided in two parts, coding of the potential phenotype and making it grow to its maximum extension.

The initial approach tested is basically equivalent to the procedure introduced in [O'Neill 2001] with the addition of environment adaptation elements, which greatly enhance the effectiveness of the process.

The genetic code is formed by a sequence of 8-bit integer numbers (codons) which control the selection of the production rules by a simple transcription process (\( n \mod m \)) where \( n \) is the integer that represents the codon (int 0..255) and \( m \) is the number of choices for the appropriate production rule. The result of this operation is mapped to the corresponding rule according to the matching number in square brackets. If for a particular rule there is only one choice, it is immediately expanded without consuming any element of the genotype.

4.1. A BNF for drawings

Any BNF grammar is a tuple \( \{N,T,P,S\} \), where \( N \) is the set of non-terminals, \( T \) is the set of terminals, \( P \) is the set of production rules mapping \( N \) to \( \{N,T\}^* \), and \( S \) is a member of \( N \) acting as the start symbol. In the proposed BNF for the coding of geometric figures:

\[
N=\{\text{expr}, \text{orig}, \text{struct}, \text{form}, \text{shape1}, \text{shape2}, \text{angle}, \text{close}, \text{len}\} \\
T=\{0,1,2,3,4,5,6,7,C,L,\text{,/, },\text{\textbackslash},\text{,},\text{,},,\text{,},\text{,}@,\Omega,\pi,\text{int}\} \\
S=\text{expr}
\]

And the production rules \( P \) are:

(1) \( \text{expr} ::= \text{orig}<\text{struct} \)
Where C represents a sector of angle degrees of an ellipse with eccentricity len; angle is a quantized value in increments of 45°; \(<\), L, and \(\backslash\) represent two lines at 45°, 90° and 135° respectively; and 4⁰, 8⁰ and π are 3 parametric forms representing their corresponding shapes. This simple BNF makes it possible to build a large number of shapes. For example, the codings:

1. \(x0\ y0\ 0\ L\ d1\ d2\ I\)
2. \(x0\ y0\ 2\ C\ d1\ I\)
3. \(x0\ y0\ 3\ L\ d1\ d2\ L\)
4. \(x0\ y0\ 0\ \backslash\ d1\ d2\ 7\ L\ d1\ d1\ L\ /\)
5. \(x0\ y0\ 4\ L\ d1\ d2\ 4\ <\ d3\ d4\ I\)
6. \(x0\ y0\ 4\ L\ d1\ d2\ 4\ <\ d3\ d4\ L\)
7. \(x0\ y0\ 4\ L\ d1\ d2\ 4\ <\ d3\ d4\ S\)

where:
- d1, d2... represent different values each time.
- \(L\) closure (e.g. 6) implies a random choice of \(r\).

Following a clockwise order, as shown in Fig 1, the examples generate the drawings show in Fig 2.

![Fig 1](image)

Notice that, while the translation from genes to drawing (phenotype) is deterministic, the same figure can be described by more than one code. For instance, example 4 can also be described by

\(x0\ y0\ 1\ \backslash\ d1\ d2\ \backslash\ d1\ d1\ L\ /\)

This is not a problem; on the contrary, it is another type of genetic degeneracy which increases the resilience of the genetic variety of the population.

For our initial experiments we have introduced further simplifications to reduce the search space. The simplified version of the BNF becomes:

(1) \(<\text{expr}> ::= <\text{len}> <\text{angle}> <\text{shape}> <\text{len}> <\text{close}>\)
(2) \(<\text{angle}> ::= 0 | 1 | 2 | \ldots | 7 \quad [0-7]\)
(3) \(<\text{shape}> ::= \text{L} | \text{\backslash} | \text{I} \quad [0-3]\)
(4) \(<\text{len}> ::= \text{int} | \text{int} \in [\text{min}, \text{max}]\)
(5) \(<\text{close}> ::= \text{I} | \text{L} \quad [0-1]\)

Therefore the simplified rule set has the following choices:

<table>
<thead>
<tr>
<th>Rule</th>
<th>Choices</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>8</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>4</td>
<td>N</td>
</tr>
<tr>
<td>5</td>
<td>2</td>
</tr>
</tbody>
</table>

In this case, our genotype is always 7-codon long and can be converted with the choices vector \(0\ 0\ 8\ 2\ 0\ 0\ 2\), where each integer correspond to the previous variable \(m\) used in the modulo operation, except when \(m=0\), in which case the translation is based on

\[\text{min} + \text{n mod} [\text{max} - \text{min}]\]

where \text{min} and \text{max} are the minimum and maximum values allowed for the length dimensions.

![Fig 2](image)

4.2. Reduction of search space table

In a square plot of side \(n\), the search space grows with \(2^n\), where \(N=n^2\). A quick estimation of the number of valid individuals (those allowed by the BNF) gives \(3^n n^2\), while the actual search space, using the simplified version of the
BNF, grows with 20.5^n^4 if the maximum building surface is limited to 80% of the plot. As shown in table 1, although the introduction of a BNF has reduced enormously the search space, the likelihood of generating a valid individual is still very small, about 1.5% for a 10x10 plot. And we are talking of a toy example. At least a 50x50 plot should be used for any practical use.

<table>
<thead>
<tr>
<th>n</th>
<th>Sq. size</th>
<th>Abs. Max.</th>
<th>3^n^3</th>
<th>BNF space</th>
<th>Viable indiv.</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>25</td>
<td>3,4E+07</td>
<td>375</td>
<td>12.800</td>
<td>2.9%</td>
</tr>
<tr>
<td>10</td>
<td>100</td>
<td>1,3E+30</td>
<td>3,000</td>
<td>204,800</td>
<td>1.5%</td>
</tr>
<tr>
<td>20</td>
<td>400</td>
<td>2,6E+12</td>
<td>24,000</td>
<td>3,3E+06</td>
<td>0.7%</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>8,5E+27</td>
<td></td>
</tr>
<tr>
<td>30</td>
<td>900</td>
<td>0</td>
<td>81,000</td>
<td>1,7E+07</td>
<td>0.5%</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>3,7E+75</td>
<td></td>
</tr>
<tr>
<td>50</td>
<td>2,500</td>
<td>3</td>
<td>375,000</td>
<td>1,3E+08</td>
<td>0.3%</td>
</tr>
</tbody>
</table>

Table 1. Search Spaces

The table shows that in GA a binary coding of length L produces a search space of size 2^L, while in GE this is reduced to \( \Pi_{i=1}^{L} C_{i} \), where \( C_{i} \) is the number of choices for rule i. Choosing wisely a BNF can reduce enormously the search space. To get some feeling for these numbers, remember that a classical AI book [Winston 1992] compares \( 2^{400} \) to the number of chessboard configurations in a game of 100 moves, defining it as a *ridiculously large number*, even compared with the number of atoms in the universe (estimated to be less than \( 2^{100} \)).

However, there is still a huge room for improvements. The two main factors producing this still too large number are the start position and the two lengths of the L or < forms. We propose a number of solutions and will experiment with them shortly.

4.3. Fitness function rationale

The obvious fitness function for this problem computes the area of the plan defined by the genome, subtracting some penalty for felling trees. The question is what should be the penalty. Let’s assume there is a single tree in a more or less central location, as in Fig 3.

![Fig 3](image)

The fitness function has to make it unworthy to fell a single tree, otherwise the obvious solution of using the whole plot (or the maximum allowable surface) would be a better choice than avoiding felling the tree. This means that using about half the surface has to be a better choice than using the whole and felling the tree.

The penalty for felling cannot be a fixed value, because, as the surface increases, the penalty per tree felled would diminish. We came to a heuristic penalty of \( 2.5^n \) where n is the number of trees affected. To further reinforce a sharp fit to the maximum available surface, a tree fully inside the selected area counts as 1, but if it is on the wall it counts only as 0.5. This is justified by the fact that a tree on the wall can be spared at an acceptable cost (by using an omega shape around it, as in the complete BNF).

5. Experiments and Results

We have run a number of trials with the simplified BNF for a 10x10 plot, using generations of 32 individuals and limiting the number of cycles to 200000 or 300000. This number is unacceptably large for a practical solution, but it is appropriate for our initial analysis. In the worst case, we could have generated 16 times each possible outcome from the BNF. In this case, it would be better to sequentially try all the possible individuals. Of course, this is not so, for crossover does not search the whole space, only the areas around the best individuals which chance created. But in most cases an acceptable solution was found much earlier. Mutation provides a broader exploration, but this genetic operator has a very small likelihood of creating a valid individual: 98.5% of the mutations are just thrown away. This also happens with the randomly generated first generation. It is quite remarkable that even with such small chances the system is able to find a good solution in a relatively small number of cycles.

In each generation, the best half of the population is kept, while the other half is replaced by new individuals, obtained from random coupling amongst the selected parents. To avoid the excess of reproduction of the fittest, which results from a roulette based coupling, we limit offspring to a couple of children for each couple of parents. Offspring from twin parents is also forbidden, forcing mutation when there is no other possible coupling available.

Results show what some authors have pointed out before: as long as there is genetic diversity, and under certain circumstances, the efficiency of crossover is much higher than that of mutation [Syswerda 1989], although there is no general agreement on that [Spears 2000]. In any case, once the genetic diversity has been lost, mutation is the only way out. This is, however, very time consuming.

Table 2 summarizes 26 runs of our experiment. The last two lines show the unweighted averages of the total set of runs and of the subset formed by runs 1-16. 62% of the runs found the best solution in less than 30,000 cycles (an average of 94 seconds in a 2005 desktop PC).
Even more interesting was to notice that a second best solution was found in 92% of the cases in less than 5,000 cycles (or 8 seconds). This is a noticeable result, compared with those found in the research for comparable search spaces, like in [Spears 2000] and others. This can be reduced enormously by improving the genotype-phenotype translation.

On the one side, it has been shown that only 1.5% of the generated individuals are valid (this is reduced to 0.7% in a 20x20 plot). In addition, we observe a very good convergence to very good results in a few cycles. We propose to modify the GE translation process to take the environment into account, either by reducing the too large individuals or by growing the small ones, which in both cases are discarded in the current approach.

### Table 2

<table>
<thead>
<tr>
<th>Num.</th>
<th>Gener.</th>
<th>t(s)</th>
<th>Gen/s</th>
<th>Num.</th>
<th>Gener.</th>
<th>t(s)</th>
<th>Gen/s</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>3</td>
<td>0.03</td>
<td>100.0</td>
<td>1</td>
<td>2</td>
<td>0.02</td>
<td>100.0</td>
</tr>
<tr>
<td>2</td>
<td>56</td>
<td>0.8</td>
<td>74.7</td>
<td>14</td>
<td>0.04</td>
<td>316.3</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>75</td>
<td>1.0</td>
<td>75.0</td>
<td>2</td>
<td>16</td>
<td>0.2</td>
<td>74.7</td>
</tr>
<tr>
<td>4</td>
<td>190</td>
<td>0.6</td>
<td>306.5</td>
<td>15</td>
<td>16</td>
<td>0.2</td>
<td>78.7</td>
</tr>
<tr>
<td>5</td>
<td>1124</td>
<td>15.3</td>
<td>73.3</td>
<td>4</td>
<td>18</td>
<td>0.06</td>
<td>306.5</td>
</tr>
<tr>
<td>6</td>
<td>1803</td>
<td>24.0</td>
<td>75.0</td>
<td>3</td>
<td>23</td>
<td>0.3</td>
<td>75.0</td>
</tr>
<tr>
<td>7</td>
<td>2074</td>
<td>27.3</td>
<td>76.0</td>
<td>13</td>
<td>32</td>
<td>0.4</td>
<td>76.2</td>
</tr>
<tr>
<td>8</td>
<td>3128</td>
<td>9.9</td>
<td>316.3</td>
<td>7</td>
<td>37</td>
<td>0.5</td>
<td>76.0</td>
</tr>
<tr>
<td>9</td>
<td>3925</td>
<td>12.8</td>
<td>306.9</td>
<td>25</td>
<td>44</td>
<td>0.6</td>
<td>75.4</td>
</tr>
<tr>
<td>10</td>
<td>5415</td>
<td>71.3</td>
<td>75.9</td>
<td>12</td>
<td>60</td>
<td>0.8</td>
<td>72.4</td>
</tr>
<tr>
<td>11</td>
<td>13412</td>
<td>179.6</td>
<td>74.7</td>
<td>5</td>
<td>77</td>
<td>1.1</td>
<td>73.3</td>
</tr>
<tr>
<td>12</td>
<td>14415</td>
<td>199.2</td>
<td>72.4</td>
<td>10</td>
<td>123</td>
<td>1.6</td>
<td>75.9</td>
</tr>
<tr>
<td>13</td>
<td>15283</td>
<td>200.6</td>
<td>76.2</td>
<td>14</td>
<td>227</td>
<td>0.9</td>
<td>263.4</td>
</tr>
<tr>
<td>14</td>
<td>15451</td>
<td>58.7</td>
<td>263.4</td>
<td>16</td>
<td>308</td>
<td>4.2</td>
<td>74.1</td>
</tr>
<tr>
<td>15</td>
<td>24409</td>
<td>310.3</td>
<td>78.7</td>
<td>26</td>
<td>464</td>
<td>6.5</td>
<td>71.0</td>
</tr>
<tr>
<td>16</td>
<td>29089</td>
<td>392.8</td>
<td>74.1</td>
<td>9</td>
<td>512</td>
<td>1.7</td>
<td>306.9</td>
</tr>
<tr>
<td>17</td>
<td>4143</td>
<td>117.9</td>
<td>348.9</td>
<td>20</td>
<td>526</td>
<td>6.8</td>
<td>76.9</td>
</tr>
<tr>
<td>18</td>
<td>67153</td>
<td>239.2</td>
<td>280.7</td>
<td>21</td>
<td>706</td>
<td>9.5</td>
<td>74.3</td>
</tr>
<tr>
<td>19</td>
<td>97723</td>
<td>1339.3</td>
<td>73.0</td>
<td>6</td>
<td>838</td>
<td>11.2</td>
<td>75.0</td>
</tr>
<tr>
<td>20</td>
<td>125077</td>
<td>1626.5</td>
<td>76.9</td>
<td>23</td>
<td>1460</td>
<td>18.9</td>
<td>77.1</td>
</tr>
<tr>
<td>21</td>
<td>200000</td>
<td>2692.6</td>
<td>74.3</td>
<td>17</td>
<td>2810</td>
<td>8.1</td>
<td>348.9</td>
</tr>
<tr>
<td>22</td>
<td>200000</td>
<td>2656.1</td>
<td>75.3</td>
<td>11</td>
<td>3455</td>
<td>46.3</td>
<td>74.7</td>
</tr>
<tr>
<td>23</td>
<td>200000</td>
<td>2594.3</td>
<td>77.1</td>
<td>24</td>
<td>4323</td>
<td>59.0</td>
<td>73.3</td>
</tr>
<tr>
<td>24</td>
<td>200000</td>
<td>2730.3</td>
<td>75.3</td>
<td>18</td>
<td>4337</td>
<td>15.4</td>
<td>280.7</td>
</tr>
<tr>
<td>25</td>
<td>232954</td>
<td>3091.5</td>
<td>75.4</td>
<td>22</td>
<td>5297</td>
<td>70.3</td>
<td>75.3</td>
</tr>
<tr>
<td>26</td>
<td>300000</td>
<td>4222.6</td>
<td>71.0</td>
<td>19</td>
<td>8234</td>
<td>8.73</td>
<td>132</td>
</tr>
</tbody>
</table>

| 68996 | 877.5 | 128.6 | 1306 | 1.45  | 128.6 |

62% 8116 94.0 132.4 92% 851 8 133 S

### 6. CONCLUSIONS & FUTURE WORK

The problem of drawing the plan of a building subject to requirements and constraints has been solved using Grammatical Evolution. The system finds good solutions quickly, but it may take much longer to find the best possible one.

Crossover seems to be a much more efficient genetic operator procedure than mutation, as long as there is sufficient genetic diversity. As a corollary, it is very important to have a good initial population.

Blind and random search could be improved by combining this technique with others which exploit the adaptation to the environment.

At least three possible lines of improvement have been identified and will be explored.

### 7. REFERENCES

EVALUATION OF SIMULATIONS WITH CONFLICTING GOALS WITH APPLICATION TO CLEANING OF YOUNG FOREST STANDS

Ulla Ahonen-Jonnarth* and Jan Odelstad*
Department of Mathematics, Natural and Computer Sciences
University of Gävle
Sweden
E-mail: uah@hig.se, jod@hig.se

*Names in alphabetical order

KEYWORDS
Evaluation, conflicting goals, agent-based simulation, cleaning of forests.

ABSTRACT
The evaluation of an activity that includes conflicting goals is a difficult task. In this methodological paper we present at first a formal structure of activities and their evaluations. This structure is then applied to the activity of cleaning young forests, which is chosen as an illustration of an activity which contains conflicting goals. An additive utility function for evaluating the cleaning of forest stands is presented. Three tree specific parameters and three global forest stand parameters were used for calculations of the utility values. The evaluation function was tested on some simulated forest stands. The approach used illustrates the usefulness of this kind of evaluation and a more detailed evaluation function could be applied for example in creating and testing cleaning algorithms.

INTRODUCTION
The forest industry is a significant economic factor in many parts of the world and, consequently, forest management treatments are important activities in several countries. The efforts to construct means for facilitating work in forests have a long tradition and there are nowadays many attempts to develop computer aided equipments. In the long perspective a goal seems to be that autonomous robots will be doing a substantial part of the work in forests. There are different kinds of forest management treatments, for example cleaning, thinning and harvesting (about an algorithm for cleaning, see Vestlund et al. 2005 and about algorithms for thinning and harvesting, see Söderbergh and Ledermann 2003). The automatization of such activities requires methods for deciding which trees shall be taken away and which will be left standing. Such decision methods should function on-line and must of course lead to good results. It is therefore important to have a method for post-factum evaluation of the results when applying different on-line decision methods. The method for post-factum evaluation of an activity shall of course be determined with respect to the goals of the activity and measure its appropriateness. Since the goals can be conflicting the measure of appropriateness has to be able to determine a trade-off between different goals. To study the

interplay between on-line decision methods and post-factum evaluation methods, simulation techniques can be used. The applications of different on-line decision methods are simulated and the results evaluated and compared. In this paper we present a rather general formal model of an activity and its evaluation (the activity involving a sequence of decisions to be taken) and discuss more specifically cleaning of young forest stands as a concrete application. The purpose of our paper is to contribute to the methodology of evaluating decision methods using simulations.

THE FORMAL STRUCTURE OF AN ACTIVITY
An activity is based on actions performed by one or more agents. Accordingly, one can distinguish between one-agent-activities and multi-agent-activities. The execution of a specific instance of an activity involves the performance of a number of actions, often appropriately represented as a sequence of actions. Let $V$ be an activity and $c$ an instance of $V$. The execution of the instance $c$ of the activity $V$ starts from an input and results in an output. Let $c'$ be the input of $c$ and $c''$ the output of $c$. The input of $c$ is often an initial state while the output of $c$ is often a set of possible final states. If the output of an instance of the activity $V$ is always singleton, then the execution of $V$ is deterministic. If $c$ is an instance of a deterministic activity $V$ and $c'' = \{s\}$ we will often, for the sake of simplicity, say that $c'' = s$, i.e. we identify the output with the element in the singleton set.

Suppose that $V$ is an activity. There may be different ways of (or different procedures for) executing instances of the activity $V$ and each way or procedure is represented by a function $F$ such that $F(c') = c''$. If $F$ and $G$ are different ways of executing $V$ we say that $F$ and $G$ are extensionally equivalent, which is denoted $F =_G$, if $F(c') = G(c')$ for all instances $c$ of $V$.

There are different modes of definition of $F$ and depending on the mode of definition there are different kinds of questions to ask. We will here give three examples.

(I) $F$ is determined as the agent $w$’s way of executing the activity $V$. This presupposes that the agent is sufficiently "reliable" such that one can suppose that the agent has a
procedure for executing the activity \( V \). In this case one can be interested in

(i) making explicit the rules that \( \omega \) uses, such that every competent agent can determine \( F \) ’s value for an argument \( c' \) by applying the rules, or

(ii) determining a computational function \( G \) which is extensionally equivalent to \( F \).

(II) \( F \) is defined by a rule of systems such that an agent system \( A \) which complies with the rules "computes" \( F \). In this case one may want to characterize \( F \) as a computationally more effective function that is not necessarily agent-based.

(III) \( F \) is defined as a computational function, in which case it can be of interest to determine a system of rules such that if an agent follows the rule the result will be \( F(c') \) for all instances \( c \) of \( V \).

Forest cleaning as a kind of activity

It does not seem to be adequate to consider 'forest cleaning' as an activity - it is too broad a concept and is more properly classified as a *kind of activity*. However, the cleaning of a special kind or type of forest for a certain aim or purpose is an activity. An instance of a cleaning activity is the cleaning (specified in an appropriate way) of a stand. Input of the instance is the stand before cleaning and the output is (a) the stand after cleaning if the activity is deterministic and (b) the set of possible results of cleaning the stand if the cleaning activity is indeterministic. A cleaning activity is often executed by one agent alone and cleaning is therefore in many contexts a one-agent-activity.

Let \( \mathcal{R} \) denote the activity 'cleaning of the kind \( \kappa \) of forests for the purpose \( \pi' \). \( \kappa \) can, for example, be a 'mixed forest' or a 'deciduous forest'. \( \pi \) can be related to the intended use of the forest in the future. An instance \( c \) of this activity is the cleaning of a certain stand \( p \). The input of the instance, i.e. \( c' \), the stand \( p \) before it is cleaned and we denote it \( p_0 \).

This means that \( c' = p_0 \). The output of \( c \) is the stand \( p \) after the cleaning has been done. Agents executing \( \mathcal{R} \) are today human beings. A human cleaner who is going to execute the instance \( c \) does not need a description of \( p_0 \), but an artificial cleaner needs a description or representation of the input of \( c \). Such a representation may consist, for example, of a description of each tree in the stand, the position of the trees, the topography of the land, the quality of the soil, the existence of obstacles (for example large stones and ditches) and the climate (micro as well as large scale) of the stand.

In Vestlund et al. (2005) a way of cleaning young forest stands is described as an on-line algorithm expressed in an informal language. The algorithm is at least partially based on interviews with professional cleaners. Vestlund et al. (2005) have also implemented the algorithm in a programming language and in *silico* executed cleaning of forest stands represented by computer files.

For testing different ways of executing cleaning, it is convenient to use a platform for simulations of young forest stands. We have built such a platform (for technical reports, see Larsson et al. 2004, Ahonen-Jonkarth and Odelstad 2005). The platform is used for two different tasks, 1) to create replicate forest stands and 2) to perform cleaning on field data or on simulated data. The objective behind the first task is to create replicates of forest stands that belong to certain forest types. The basic data for a replicate simulation can be field data or simulated data. Data of a basic forest type is used for constructing a decision tree structure and probabilities are calculated for different parameter values of a tree, such as diameter, eventual damage and species. A user can also apply her own principles of forest stand parameters and thus decide more or less strictly how the resulting forest is going to be. A result of a simulation, a young pine forest stand, is shown in Figure 2.

![Figure 1](image1.png)

Figure 1. A schematic example about cleaning of forest stand A. The cleaning method \( \mathcal{R}_1 \) gives a different result (\( B_{R1} \)) than the cleaning method \( \mathcal{R}_2 \) which gives the result \( B_{R2} \).

![Figure 2](image2.png)

Figure 2. A young pine forest stand that was simulated by using our simulation platform. The larger the size of a dot, the larger the diameter of the tree the dot represents. Dark dots represent conifer trees (mainly pines) and the light dots represents birch trees.
We have also made an implementation (at least an approximate one) of the cleaning algorithm presented in Vestlund et al. (2005), and we can use this for simulation of cleanings of our simulated stands.

**EVALUATING ACTIVITY PROCEDURES**

Different procedures for executing an activity V can be more or less good (appropriate). It is therefore important in many contexts to evaluate different ways of executing activities. Suppose that F and G are different ways of executing V. It is important to note the difference between how good F is as a way of executing a given instance c of V and how good F is a way of executing V in general. It is thus important to distinguish between

- \( P(c) \succ V G(c) \), i.e. the execution of the instance c of V by F is better than by G.
- \( F \triangleright V G \), i.e. F is quite generally a better way of executing V than G.

In certain cases there exists a measure of how good different procedures for executing an instance c of the activity V is and a measure \( U_V \) of how good different procedures for the activity V is generally. It is reasonable to suppose that the following holds:

- \( u_F(F(c)) = u_G(G(c)) \) iff \( F(c) \succ V G(c) \).
- \( U_F(F) = U_G(G) \) iff \( F \triangleright V G \).

If for all instances c of an activity V it holds that \( F(c) \succ V G(c) \) then it seems uncontroversial to conclude that \( F \triangleright V G \).

But if for some instances c it holds that \( F(c) \succ V G(c) \) while for other instances c it holds that \( G(c) \succ V F(c) \) then it is more difficult, in many cases impossible, to decide if \( F \triangleright V G \) or not.

The binary relations \( \succ V \) and \( \triangleright V \) and the measures \( u_V \) and \( U_V \) concern how good different execution procedures are (for given instances or generally) "all things considered". These relations and measures can be difficult to establish since goodness (appropriateness), all things considered, is in many contexts a very complex attribute. However, goodness (appropriateness) with respect to a certain aspect (attribute) \( \alpha \) may be easier to ascertain. Let

- \( F(c) \succ V \alpha G(c) \) denote that F is a better way with respect to \( \alpha \) than G to execute the instance c of V;
- \( F \triangleright V \alpha G \), denote that F is a better way with respect to \( \alpha \) than G to execute V.

\( u_F(F(c)) \) is a measure of how good F is with respect to \( \alpha \) as a way of executing the instance c of the activity V.

\( U_F(F) \) is a measure of how good F is generally with respect to \( \alpha \) as a way of executing the activity V.

\( u_V \) and \( U_V \) can be considered as a kind of utility function. Utility functions are largely used in economics. In addition, utility functions play an important role in the study of intelligent agents within the discipline of artificial intelligence, as the following quotation from Russell & Norvig (2003, p. 52) emphasizes: "... when there are conflicting goals, only some of which can be achieved ..., the utility function specifies the appropriate tradeoffs. ... when there are several goals that the agent can aim for, none of which can be achieved with certainty, utility provides a way in which the likelihood of success can be weighed up against the importance of the goals."

Suppose that different procedures for executing V is evaluated with respect to the attributes \( \alpha_1, ..., \alpha_k \). Then

\[
\begin{align*}
    u_V(F(c)) &= \prod_{\alpha} u_{V, \alpha}(F(c))
\end{align*}
\]

Then the form of the function f can differ considerably in different contexts. But it is often a desideratum that f has a simple form, for example being additive or multiplicative.

For many aspects \( \alpha \) of importance for how good F is as a way of executing V, but certainly not for all aspects, it seems reasonable that the following holds:

- \( F(c) \succ V \alpha G(c) \Rightarrow F(c) \succ V G(c) \) ceteris paribus (everything else held constant)
- \( F \triangleright V \alpha G \Rightarrow F \triangleright V G \) ceteris paribus (everything else held constant)

It is of course desirable to have a complete method for evaluating different ways of executing an activity but when this is not possible the concepts introduced above can still make a partial evaluation possible, and this may in certain situations be of great interest.

Suppose that the relation \( \triangleright V \) is defined for the activity V. Then an optimal way of executing V is a procedure F that is maximal with respect to \( \triangleright V \). It is usually desirable to choose an optimal way of executing F, but it can be difficult to find such a way if the number of different ways to execute V is large. The search for an optimal way of executing V may then be replaced by a more restricted goal, viz. the search for an acceptable way, i.e. a way that is good enough (sufficiently acceptable).

**EVALUATIONS OF CLEANING**

We need principles for evaluating how good different cleaning procedures are. Today there seems to be no consensus regarding principles which permit a complete evaluation. But results from different procedures for \( \mathcal{R} \) can in certain cases be compared with respect to \( \triangleright \). One of the problems with characterizing good cleaning is that there seems to be few studies showing the result of cleaning over a long period of time (several decades). Hence, only
principles for partial evaluations seem to be at hand for the moment.

In this paper we use two kinds of parameters for the calculation of utility values of a cleaned stand: tree parameters and global stand parameters. The tree parameters are attributes of individual trees. For calculations of stand parameters global information of a stand is used, either by itself or together with some individual tree attribute values. The model we use is quite simple but our goal here is to test the methodology which can later be used for making the evaluation function more complex.

We have here chosen an additive function to be used for evaluation of the cleaning results, see equation (1). The function is based on tree parameters and global stand parameters. The utility $u$ for each cleaned stand $B^c$ depends on the parameters of each tree $t$ in $B^c$ and the parameters of the stand $B^c$ itself. The utility of the tree attribute $a_t$ is represented by $u^a_t$ and utilities for stand attribute $a_s$ is represented by $u^s$.

$$ u(B^c) = \sum_{t \in B^c} \left( \sum_{i=1}^{n} u^a_i(t) \right) + \sum_{j=1}^{m} u^s_j(B^c) $$

There are some general goals for good cleaning that are often acceptable taken one by one. Vestlund interviewed thirteen professional cleaners who stated that “a preferable stem should be of desired species, straight, healthy, and have a preferable size and a suitable position in comparison to surrounding stems” (Vestlund 2004). Furthermore, Vestlund points out that “Allowed percentages of deciduous stems in coniferous stands vary from about 5 to 20% for different companies and in different parts of Sweden” (Vestlund 2004). However, together these goals create a complex system where different goals are dependent on each other and in several cases conflict.

We will in this paper consider the following as general goals for cleaning of young forests.

A. Increase the mean diameter of trees.
B. Increase the proportion of undamaged trees.
C. Obtain a certain proportion of birches.
D. Obtain as even distribution of trees as possible.
E. Obtain a certain amount of stems per area (a certain density).

A cleaner must take all these goals into consideration during a cleaning process. In many cases the goals are in conflict with each other. For example goals A and B could be formulated in following ways.

A’. Choose trees having a diameter near the mean diameter rather than trees that are very small or very big.

B’. Choose healthy trees prior to damaged ones.

Both goals are easy to fulfill ceteris paribus but it is not common that only one parameter varies.

In the experiment described below the utilities for the tree parameters are $a_1$ diameter-utility, $a_2$ health-utility (actually undamaged-utility or non-visible-damage utility) and $a_3$ proximity-utility. The diameter-utility for a tree $i$ is composed of the diameter multiplied with a scaling constant:

$$ u^a_i(t) = k_1 \cdot d_i $$

where $d_i$ is the diameter of the tree $t_i$ and $k_1$ is a scaling constant for this parameter.

The value of the health-utility for a tree is a positive number if the tree is undamaged and a negative number if the tree is damaged. The numbers are multiplied with a scaling constant $k_2$.

Proximity-utility is defined as follows:

- the value of a tree is zero if there are no trees inside an area $A$ around the tree
- the value of a tree is negative (penalty value) if there is at least one tree inside the area $A$. The value depends on the number of trees inside the area $A$; the penalty value is larger if there are more trees inside the area $A$.

Global information is used for calculations of stand utilities. The stand utilities used here are the utility determined by the variation around the mean diameter, the utility of proportion of birches in the stand and the utility of proximity of other trees. For all these stand parameters, intervals are used when amount of utility values are assigned. For the diameter parameter each tree gets a utility value depending on how close the diameter of the tree is to the mean diameter of the stand.

**EXPERIMENTS**

In order to test this evaluation model we used simple manipulations and computer simulations. The simulations were performed using field data from Vestlund (2001) concerning a young pine forest stand. Information of different tree parameters (diameter on breast height, eventual damage and species) were extracted from the field data and probabilities for each combination of attribute values were saved in a decision tree-diagram. This decision tree-diagram was then used for creating replicate forest stands that represent the same forest type as the stand from which the data was collected. The simulated forest stands were then cleaned using the cleaning algorithm implemented in our simulation platform. The algorithm is based on the principles presented in Vestlund et al. (2005). Vestlund’s approach is basically a procedural one choosing and saving trees in two steps according to certain rules about diameter, health, species and number of trees saved earlier. We also created test forests, i.e. an ideal forest that fulfills the earlier presented goals. We manipulated forest stands further by removing trees and planting additional trees instead and could thus investigate the behaviour of the evaluation function and its dependence on different
parameters. Utility values were then calculated for all these manipulated forest stands as well as the simulated forest stands. The numeric values and the intervals used for the evaluation function are presented in Table 1. In subsequent studies we are planning to develop the evaluation function further and use it in generating other cleaning algorithms, for example by using AI methods.

Table 1: The parameter values, scaling constants of tree parameters and intervals together with respective utility values of stand parameters used for experiments with the evaluation function.

<table>
<thead>
<tr>
<th>Tree parameters</th>
<th>Scaling constant</th>
<th>Value of scaling constant</th>
</tr>
</thead>
<tbody>
<tr>
<td>Diameter</td>
<td>$k_1$</td>
<td>0.1</td>
</tr>
<tr>
<td>Health</td>
<td>$k_2$</td>
<td>0.3</td>
</tr>
<tr>
<td>Proximity</td>
<td>$k_3$</td>
<td>0.5</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Stand parameters</th>
<th>Interval</th>
<th>Utility value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Diameter vs.</td>
<td>$-0.4^*d_m \leq d_i \leq 0.4^*d_m$</td>
<td>1</td>
</tr>
<tr>
<td>mean diameter</td>
<td>$-0.6^*d_m \leq d_i &lt; -0.4^*d_m$</td>
<td>-1</td>
</tr>
<tr>
<td></td>
<td>$0.4^*d_m &lt; d_i \leq 0.6^*d_m$</td>
<td>-2</td>
</tr>
<tr>
<td>Proportion</td>
<td>$8% \leq b_p \leq 12%$</td>
<td>5.0</td>
</tr>
<tr>
<td>birches</td>
<td>$12% &lt; b_p$</td>
<td>-4.0</td>
</tr>
<tr>
<td></td>
<td>$b_p \leq 8%$</td>
<td>-5.1</td>
</tr>
<tr>
<td>Mean density</td>
<td>$2400 \leq d_e \leq 3000$</td>
<td>5.0</td>
</tr>
<tr>
<td></td>
<td>$d_e &lt; 3000$</td>
<td>-4.9</td>
</tr>
<tr>
<td></td>
<td>$2400 &lt; d_e$</td>
<td>-5.1</td>
</tr>
</tbody>
</table>

$\quad d_m$ – mean diameter, $d_i$ – diameter for a tree $i$
$b_p$ – proportion of birches in a stand
$de$ – density value (stems per hectare)

For testing the evaluation function the following experiments were performed using simulated or manipulated forest stands.

A) An ideal forest stand was created including 36 trees that correspond to a density of 2500 stems per hectare (see Figure 3). This stand was further manipulated by adding 5, 9 or 12 ideal trees or simulated trees or by removing 3 trees. Utility values were calculated for all stands.

Experiment A1. In manipulations with added trees, the added trees were ideal ones.

Experiment A2. In manipulations with added trees, the added trees were simulated ones.

B) The simulated forest stand presented in Figure 2 was cleaned according to algorithm described above. The cleaning result is presented in Figure 4. The stand was manipulated by adding 5, 9 or 12 ideal or simulated trees or removing 3 trees. Utility values were calculated for all stands.

Experiment B1. In manipulations with added trees, the added trees were ideal ones.

Experiment B2. In manipulations with added trees, the added trees were simulated ones.

Figure 3. An example of a hypothetical ideal young pine forest stand.

Figure 4. The result of cleaning the stand presented in Figure 2.

The results of the experiments are presented in Table 2. The optimal density value is 36 trees per stand representing 2500 stems per hectare. In the experiments the forest stands having density values near the optimal value had high utility values. Mean diameter of the stands varied between 4.5 and 5.2 cm. The utility values were between 24.4 and 50.0 for the ideal forest stands and between 21.9 and 37.3 for the simulated forest stands. However, the actual utility values are not meaningful to use for comparisons between different stands but can be used when comparing the same stand after different manipulations as was done here. The utility values can be used for comparisons when the same stand is cleaned using different cleaning algorithms and thus is a tool when new algorithms are evaluated.
Table 2. Orderings of utility values for experiment A with ideal forest stand and its manipulations and experiment B with simulated forest stand and its manipulations. (The best result is denoted by 1). The optimal density in this case is 36 trees on the stand representing 2500 stems per hectare.

<table>
<thead>
<tr>
<th>Experiment A</th>
<th>Ideal forest stands</th>
</tr>
</thead>
<tbody>
<tr>
<td>36-3 (33)</td>
<td>5</td>
</tr>
<tr>
<td>36</td>
<td>1</td>
</tr>
<tr>
<td>36+5 (41)</td>
<td>2</td>
</tr>
<tr>
<td>36+9 (45)</td>
<td>4</td>
</tr>
<tr>
<td>36+12 (48)</td>
<td>3</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Experiment B</th>
<th>Simulated forest stands</th>
</tr>
</thead>
<tbody>
<tr>
<td>34-3 (31)</td>
<td>4</td>
</tr>
<tr>
<td>34</td>
<td>5</td>
</tr>
<tr>
<td>34+5 (39)</td>
<td>1</td>
</tr>
<tr>
<td>34+9 (43)</td>
<td>2</td>
</tr>
<tr>
<td>34+12 (46)</td>
<td>3</td>
</tr>
</tbody>
</table>

The highest values of the evaluation function were obtained by stands that had optimal amount of trees. These results illustrate an evaluation of an activity containing conflicting goals and the evaluation function presented here is going to be used as a basis for further development towards a more detailed evaluation function. In the present study the intervals used were quite rough and several intervals or distribution functions can be used for calculating the values of utility functions with respect to different aspects.

CONCLUSIONS

The structure of an evaluation process may be quite complex and the goals of an activity may conflict with each other. After a formal analysis of the structure of an activity and the evaluation procedure for it we conducted an application of an evaluation function for an activity with conflicting goals. Test material was obtained by simulating forest stands and further manipulations of these simulated stands. A relatively simple additive evaluation function was tested and in further work the evaluation function is going to be made more detailed. In forthcoming studies the evaluation function can be used for generating other cleaning algorithms, for example by using AI methods.

ACKNOWLEDGEMENTS

We are very grateful to Karin Vestlund who gave us the possibility to use her field data and cleaning algorithm. We would also like to thank Karin Vestlund and Märten Hugosson for interesting discussions about different aspects of cleaning. Per Larsson and Johan Lehnborn are acknowledged for technical support. Financial support was given by the Knowledge Foundation, Creative Media Lab and the University of Gävle.

REFERENCES


Modeling of Water Hammer Phenomenon in Simple Irrigation System and Comparing the Analytical to Experimental Results

Ahmad Bagheri
Majid Alitavoli
Mehdi Aghakashi
Salar Basiri
Amir Hajilooy

Dept. of Mechanical Engineering
University of Guilan, Rasht, Iran, POBox 3756
E-mail: bagheri@guilan.ac.ir

Abstract

Any variation in fluid velocity in a piping system, will produce compression waves in the system. These waves in result will produce higher pressure that the designed one inside the system which will propagate with the wave velocity. In this phenomenon which is called 'Water Hammer', factors such as opening and closing valves, starting or stopping the pump, stopping the turbine, increase or decrease in the amount of water, etc will cause water hammer.

Based on the above it is very important to model this phenomenon in order to firstly determine the high pressure points to control water hammer. To this extent, a computer program is written. The goal of this program is to compare the mathematical and analytical modeling methods with the experimental results using a device made by Bergant.

Introduction

Research in the field of Water Hammer (WH) phenomenon has a century of background. The first mathematical model developed in 1925 by professor Rich who compared its result against experimental ones [1]. Stuecker and Wylie were two pioneer researchers who spatially and completely done a survey about WH phenomenon in 1945 [2].

The importance of WH phenomenon stems from the fact that most fluid transferring systems in power stations, oil, and gas industries involves WH phenomenon. The focus of this paper is to study the modeling of WH phenomenon in water supplying system and their accessories such as pumps, turbine, valves, and reservoirs. Also an attempt was made to model the effect of WH phenomenon in an experimental apparatus build by Bergant. Then the modeling results were compared against the experimental ones done by Bergant.

WH theory

As known, variation in fluid velocity $\Delta V$ produces variation in fluid pressure head $\Delta H$. This variation in pressure will propagate by velocity $a$. In fact, as dedicated in Figure 1 the quantity of $\Delta V$ is negative, and therefore, the quantity of $\Delta H$ is positive [1, 2].

The analysis is focused on a section of pipe shown as $\delta L$ in Figure1. In this section $\delta L$ is small and considered as desired size, but not smaller than the differential $dL$ amount. Compressive wave and pipe deformation resulted by the variation in head of pressure $\Delta H$ to be propagated by velocity $a$. Instead of considering the wave velocity in ratio to water velocity, it is considered in relevance to a fixed sight on the pipe. For relatively solid pipes, selection of any coordinate reference system will give similar results.

![Figure1: The effect of WH in pipe](image)

After writing the equation of linear momentum conservative principles for mass, we can calculate $\Delta H$ in the pipe. This quantity is resulted from the following equations:

$$\Delta H = -\frac{a}{g} \Delta V \tag{1}$$

In Which:

$$a = \frac{\sqrt{K/\rho}}{\sqrt{1 + \frac{K}{E \cdot e} C}} \tag{2}$$

I this equation:

$a =$ wave velocity (speed),
$k =$ total elasticity coefficient of the medium,
$e =$ wall thickness of pipe,
$E =$ elasticity module of pipe,
$\rho =$ Fluid density

Equations for entropy of mass and momentum to describe WH modeling

As noticed, for a sudden variation of $\Delta V$ in the pipe, we can calculate the variation in the head pressure $\Delta H$. When
expanded, we can calculate the head of the pressure and its velocity for each section of pipe in any time and under imposed loading conditions. To obtain this calculation we should use equations for mass and momentum entropy [2, 3].

\[
\frac{\partial V}{\partial t} + \frac{1}{P} \frac{\partial P}{\partial s} + g \frac{dz}{ds} + \frac{f}{2D} V'V = 0
\]  
(3)

\[
a^2 \frac{\partial V}{\partial s} + \frac{1}{P} \frac{\partial P}{\partial t} = 0
\]  
(4)

In the above equations:

P = Fluid pressure

\( t \) = Time in sec

S = Pipe length

D = Internal diameter of the pipe

G = Gravity

**Extraction of definite differential equation from basic differential equation for WH**

The solving method for differential equations is a definite method. In definite methods, two partial differential equations are subsumed by one normal differential equation. This deduction starts by substituting equations 3 and 4 by several linear equations of the same type. By using \( \lambda \) as the linear coefficient (Lagrange’s coefficient), a combined linear equation is gained as follows:

\[
\lambda \left( \frac{\partial V}{\partial t} + \frac{1}{P} \frac{\partial P}{\partial s} + g \frac{dz}{ds} + \frac{f}{2D} V'V \right) + \left( \frac{a^2}{\partial s} + \frac{1}{P} \frac{\partial P}{\partial t} \right) = 0
\]

(5)

When grouped, equation 5 will convert as follows:

\[
(\lambda \frac{\partial V}{\partial t} + a^2 \frac{\partial V}{\partial s}) + \left( \frac{1}{P} \frac{\partial P}{\partial t} + \lambda \frac{\partial P}{\partial s} \right) + \lambda g \frac{dz}{ds} + \frac{f}{2D} V'V = 0
\]

(6)

If \( \lambda \frac{\partial V}{\partial t} + a^2 \frac{\partial V}{\partial s} \) is substituted by \( \lambda \frac{dV}{dt} \) then \( \lambda \frac{ds}{dt} = a^2 \), in addition if \( \frac{1}{P} \frac{\partial P}{\partial t} + \lambda \frac{\partial P}{\partial s} \) is substituted by \( \frac{1}{p} \frac{dp}{dt} \) then

\[
\lambda = \frac{1}{p} \frac{ds}{dt}
\]

(7)

From the above we obtain: \( \lambda^2 = a^2 \) so \( \lambda = \pm a \).

Now equation (6) can be rewritten once using \( \lambda = +a \) and once using \( \lambda = -a \), which leads us to a couple of the normal differential equations. By driving these equations on the velocity of wave and the variable \( p = \gamma(H - z) \), we obtain:

\[
\frac{dv}{dt} + \frac{g}{a} \frac{DH}{dt} + \frac{f}{2D} V'V = 0 \quad \frac{ds}{dt} = +a
\]

(8)

To understand the solution process we refer to Figure 2.

![Figure 2: Extraction of definite differential equation procedure](image)

Each point on the plane S-t, which is shown as point P in an independent manner, reserve to quantities like V and H. Then we draw the divertive lines \( C^P \) and \( C^F \) at point P and continue them to intersect axis S on right and left side of point P. Notice that there exist two points over S axis which their horizontal distance from P is equal to \( \Delta s \). These two points are shown as \( S_{le} \) and \( S_{ri} \). Equation (6) is used in definite extent of \( C^P \) and equation 7 is used in definite extent of \( C^F \).

**Extraction of definite algebraic equation from definite differential equation for WH**

We write the equations (7) and (8) as limited difference type [2, 3].

\[
C^P: \left( V_p - V_{ni} \right) - \frac{g}{a} \left( H_p - H_{ni} \right) + \frac{F \Delta N}{2D} \left| V_{ni}V'V \right| = 0
\]

(9)

\[
C^F: \left( V_p - V_{le} \right) + \frac{g}{a} \left( H_p - H_{le} \right) + \frac{F \Delta N}{2D} \left| V_{le}V'V \right| = 0
\]

(10)

Also by writing the definite equations as the limited difference we have:

\[
\Delta s = \pm a \Delta t
\]

(11)

Now we solve the above equations by numerical limited difference method. First, we should decide to divide the pipe into several sections in the direction of S axis. If \( \Delta N \) is the number of division, then we have \( \Delta s = \frac{L}{\Delta N} \). Based on the amount of \( \Delta s \) from equation 11 we can calculate \( \Delta t \). Now we can make a network of definitive shown in Figure 3.
The points shown are selected to be in the direction of pipe with distance of $\Delta s$. The quantities $V$ and $H$ at these points are in primary conditions. Normally the primary conditions are part of $V$ and $H$ which is obtained from a stable flow condition at the start of a passing made. With the quantities $L_0$ and $F_i$ we can solve equations 9 and 10 simultaneously. Then we can obtain quantities $P_{u,v}$ and $V_{u,v}$ from two points at period $t = \Delta t$. Boundary conditions at $(S = 0)$ and $(S = L)$ is used for calculating $H_{p,v}$, $H_{p,f}$. Then all quantities for $V$ and $H$ in period $t = 2\Delta t$ by using the calculated quantities in the period $t = \Delta t$ as known quantities is earned. This process is used repeatedly to obtain $V_{N+1}$, $H_{N+1}$.

Analysis for the separation of relieved water and air columns

The separation of fluid column is occurred when the pressure of fluid drops below saturation pressure of steam. The cross sign describes the separation phenomenon. Firstly we should prepare a model [7].

The simplest model for the separation, neglects the existence of unsaturated gases which may exit in pressures below the fluid pressure. Instead, it is assumed that fluid is continuously connected up to the steam pressure limit. After this point, bubble growth in constant pressure continues equally to steam pressure. Also this simple model requires assumptions about forming of bubble. Actually the process of forming bubble is very complex and hence the simulation is almost impossible. Therefore, we employ the simplest possible model. For this purpose, we assume that the bubble is depended to the pipe cross section and growth or death of the bubble is depended to the relational speed of bubble's wall. This survey (growth or death of bubble) need the analysis of boundary condition imposed. On the internal side of the bubble, the pressure is exactly equal to the steam flow. In each side in which separation occurs, two speeds are obtained. One speed is toward the up stream current of the bubble, and the other toward the down stream direction. Relation quantities and the direction of these speeds, determine the growth or death of bubble. This simple model is shown in Figure 4.

Figure 4: Separation of relieved water and air model

For analysis of this model the conservation of momentum is used. The collision occurred between two bubbles when moving by two different speeds. The results show an increase in the head due to the collision (Figure 4).

$$\Delta H = \frac{g}{2g} \left( V_{u,stream} - V_{d,downstream} \right)$$ (12)

This increase in head or $\Delta H$ adds to each knot until the new pressure after bubble form up is determined.

Modeling WH phenomenon in Bergant test machine

The machine made by Bergant (Figure 5) has two storage tanks, one pipe and a valve in the down side of the flow. Bergant tested WH phenomenon in different speeds of fluid and measured the pressure head in different points. Our intention was to compare Bergant's experimental results to analytical ones obtained for WH by definitive methods written by computer codes.

Figure 5: Bergant schematic

In this system, the pipe length was 37.2 m, diameter 22 mm, wave velocity of 1319 m/s and the pressure head in the storage tanks was kept constant at 22 m height and the speed in pipe was 0.2 m/s.

If the bottom valve in $t = 0.009$ is closed the WH phenomenon has been occurred in the system, as referred in references [4, 5, 6].

Results and suggestions

By studying the graphs from analytical modeling and comparing them to the Bergant results, and also results from other researches, it is observed that the accuracy and our prepared programs are very vital. As a result we can model any water network by using computer codes.
The results of the experimental jobs of Bergant have been shown in Figures 6 (for the middle of pipe) and 8 (for valves). Figures 7 and 9 shown the comparison of results for simulation, regarded for the middle of pipe and valve respectively. An acceptable compromise between two sets of results has been observed.

Figure 6: Experimental Results of water hammer in middle of pipe

Figure 7: Simulated Results of water hammer in middle of pipe

Figure 8: Experimental Results of water hammer in valve

Figure 9: Simulated Results of water hammer in valve

References:


A Spectral Expansion Method For The Simulation of Vertical Salinity Variations In Upper-Milford Haven Estuary, South Wales, UK.

Bahareh Kaveh
Naghmeh Keshavarzi-Roonizi
Rahim Ghorashi
Vahid Nassehi
Chemical Engineering Department
Loughborough University
Loughborough LE11 3TU, UK
v.nassehi@lboro.ac.uk

KEYWORDS
Spectral expansions, Salinity, Lagrangian Finite Elements

ABSTRACT
Although modelling of natural water systems is a highly established practice to obtain accurate simulations which can provide detailed description of many phenomena in such environments, requires elaborate and specialised models.

The use of spectrally expanded hydrodynamic models, which provide quantitative information for 3-dimensional systems through the solution of 2-dimensional modal governing equations, is a prime example of such an elaborate scheme. This scheme has the advantage over traditional three dimensional methods because it is computationally more cost effective. However in case of water systems such as estuaries where the flow domain geometry changes constantly during a tidal cycle, the solution of modal equations becomes extremely complicated. Therefore such models, can only be used by experts, who have considerable experience in areas of CFD.

In order to avoid the necessity of extensive training for users of sophisticated software an expert system which through the exploitation of IT techniques makes it accessible to non-experts has been designed.

INTRODUCTION
Three-dimensional modelling of hydrodynamic behaviour and pollutants dispersion in estuaries can be very costly from a computational point of view. However, in some cases it may be possible to obtain predictive simulations that provide useful quantitative information in all three spatial dimensions in a flow channel without employing a full three-dimensional scheme. Such an approach uses spectral expansion techniques to derive modal versions of governing equations. Essentially this technique replaces the solution of three-dimensional equations with alternative simulations based on the solution of a number of (usually 2 or 3 at most) two-dimensional models. Detailed derivation of modal equations for tidal wave propagation and transport equations can be found in Smith (1995). A brief outline is given in this section.

GOVERNING MODEL EQUATIONS USED IN THE PRESENT WORK

The procedure starts with transforming the three-dimensional equations to the following ‘Sigma Coordinates’ system. This system provides an efficient method for tracking the vertical motion in a tidal channel. In this system the total water depth \( H(x, y, t) \) is defined as \( H = h + \zeta \), where \( z \) the vertical co-ordinate is \( z = -h(x, y) \) at the bed and \( z = \zeta(x, y, t) \) at the free surface. We define a co-ordinate \( \sigma \) as (figure 1)

\[
\sigma = \frac{z + h}{H}, \quad 0 \leq \sigma \leq 1
\]

(1)

Figure 1. Water surface elevation with respect to a datum

Vertical velocity components at the bed and at the free surface can now be written in terms of the horizontal velocity components \( u(x, y, \sigma, t) \) and \( v(x, y, \sigma, t) \), respectively, as

\[
\begin{align*}
\dot{w} &= -u \frac{\partial h}{\partial x} - v \frac{\partial h}{\partial y} \quad \text{at} \quad \sigma = 0, \\
\dot{w} &= \frac{\partial \zeta}{\partial t} + u \frac{\partial \zeta}{\partial x} + v \frac{\partial \zeta}{\partial y} \quad \text{at} \quad \sigma = 1.
\end{align*}
\]

(2)

In the presence of a source delivering water at a rate of \( Q(x, y, \sigma', t) \), into the water column at the fractional height \( \sigma' \) above the bed, \( (Q \) measures the volume of new water into a unit volume in a unit of time) the vertical component of the velocity can be expressed as

\[
W = \frac{1}{H} \left\{ w + (1 - \sigma) \left( u \frac{\partial h}{\partial x} + v \frac{\partial h}{\partial y} - H \frac{Q d\sigma'}{\sigma'} \right) \right\}
\]

(3)
Using the Boussinesq approximation we consider a case where changes of density is small whilst changes in weight is dynamically significant. Therefore the mass conservation (continuity) equation is expressed as

\[
\frac{\partial \rho}{\partial t} + \frac{\partial}{\partial x} (\rho u) + \frac{\partial}{\partial y} (\rho v) + H \frac{\partial w}{\partial \sigma} = \int_0^1 0 \, d\sigma',
\]

(4)

This means that the variations of \( W \) effectively replaces the local water source \( Q \) by a vertically averaged source strength. The kinematic boundary conditions for no water flow across the bed \( \sigma = 0 \) and the free surface \( \sigma = 1 \): \( W = 0 \) on \( \sigma = 0, 1 \). Therefore, the free surface elevation \( \xi(x, y, t) \) is governed by the vertically integrated mass conservation equation

**Modal approach**

Consider the eddy viscosity \( \nu \) which, in general, varies approximately with fractional height above the bed. This parameter is constant for laminar flows and parabolic for turbulent flows. Let \( \hat{\nu} (\sigma) \) be a dimensionless shape function representing the idealized eddy viscosity profile in a shallow water system. The velocity mode \( \Phi^{(m)}(\sigma) \) and its associated eigenvalues \( \mu^{(m)} \) are expressed as

\[
\frac{d}{d\sigma} \left( \hat{\nu} \frac{d\Phi^{(m)}}{d\sigma} \right) + \mu^{(m)} \Phi^{(m)} = 0,
\]

(5)

Here \( \Phi^{(m)} = 0 \) on \( \sigma = 0 \); and \( \hat{\nu} \frac{d\Phi^{(m)}}{d\sigma} = 0 \) on \( \sigma = 1 \). We also have

\[
\int_0^1 \Phi^{(m)}(\sigma) \, d\sigma = 1, \quad \text{and} \quad \int_0^1 \hat{\nu} \frac{d\Phi^{(m)}}{d\sigma} \frac{d\Phi^{(n)}}{d\sigma} \, d\sigma = 0 \quad \text{for} \quad m \neq n.
\]

The velocity components are now represented by the following expansions

\[
\begin{align*}
\begin{cases}
  u = \sum_{m=0}^{\infty} u^{(m)}(x, y, \sigma) \Phi^{(m)}(\sigma), \\
  v = \sum_{m=0}^{\infty} v^{(m)}(x, y, \sigma) \Phi^{(m)}(\sigma), \\
  HW = \sum_{m=0}^{\infty} \left( \frac{\partial}{\partial x} (H\nu^{(m)}) + \frac{\partial}{\partial y} (H\nu^{(m)}) \right) \omega^{(m)}(\sigma),
\end{cases}
\end{align*}
\]

where

\[
\omega^{(m)}(\sigma) = \frac{1}{\sigma} \int_0^\sigma \Phi^{(m)}(\sigma') \, d\sigma' - (1 - \sigma) \int_0^\sigma \Phi^{(m)}(\sigma') \, d\sigma'.
\]

(7)

Substitution of expansions (7) into the continuity and momentum equations after multiplication of the original equations by \( \Phi^{(m)} \) and integration with respect to \( \sigma \) provides their corresponding modal equations. These equations are relatively lengthy relationships and since they have already been published (Smith, 1995) they are not given here.

Using a similar approach modal form of the solute transport equation can be obtained. The original solute transport equation (again using the above described \( \sigma \) coordinate system) is written as

\[
\begin{align*}
\frac{\partial}{\partial t} (Hc) + \frac{\partial}{\partial x} (Hvc) + \frac{\partial}{\partial y} (Hvc) + H \frac{\partial}{\partial \sigma} (Wc) \\
= H(q - Qc) + \sum_{j, k} \left( \frac{\partial}{\partial x} (H\nu^{(j)} \psi^{(k)}) + \frac{\partial}{\partial y} (H\nu^{(j)} \psi^{(k)}) + \frac{\partial}{\partial \sigma} (HK \nu^{(j)} \psi^{(k)}) \right)
\end{align*}
\]

(8)

Here, \( H \) is water depth, \( u, v \) and \( W \) are the longitudinal, lateral and vertical components of velocity, \( q \) and \( Q \) fresh and salt water discharges and \( K \) are the vertical and horizontal eddy diffusivity, respectively. After the derivation of the concentration modes \( \psi^{(m)}(\sigma) \) and associated eigenvalues \( \lambda^{(m)} \) via the Sturm-Liouville equation the modal expansion of the concentration variable is obtained

\[
C = \sum_{m=0}^{\infty} \psi^{(m)}(x, y, t) \psi^{(m)}(\sigma)
\]

(9)

Insertion of the modes into equation gives rise to the following modal equations which are solved to obtain the required results.

\[
\begin{align*}
\frac{\partial}{\partial t} (Hc^{(m)}) + \sum_{j, k} \left( \frac{\partial}{\partial x} (H\nu^{(j)} \psi^{(k)}) + \frac{\partial}{\partial y} (H\nu^{(j)} \psi^{(k)}) \right) \frac{1}{\sigma} \Phi^{(j)}(\sigma') \psi^{(k)}(\sigma') \, d\sigma' \\
- \sum_{j, k} \left( \frac{\partial}{\partial x} (H\nu^{(j)} \psi^{(k)}) + \frac{\partial}{\partial y} (H\nu^{(j)} \psi^{(k)}) \right) \psi^{(j)}(\sigma') \psi^{(k)}(\sigma') \, d\sigma' = H \frac{1}{q} \psi^{(m)} \, d\sigma - H \sum_{j} \psi^{(j)}(x, y, t) \psi^{(j)}(\sigma') \, d\sigma' \\
+ \sum_{j, k} \left( \frac{\partial}{\partial x} (H\nu^{(j)} \psi^{(k)}) \psi^{(j)}(\sigma') \psi^{(k)}(\sigma') \, d\sigma' + \frac{\partial}{\partial \sigma} (HK \nu^{(j)} \psi^{(k)}) \psi^{(j)}(\sigma') \psi^{(k)}(\sigma') \, d\sigma' \right).
\end{align*}
\]

(10)
Choosing value of $m$ as 0.1, etc in equation (10) various modal equations corresponding to the present solute transport model are obtained. Simulations based on this model via numerical solution of spectrally expanded equations provides information regarding flow field variables in all three spatial dimensions. The numerical method used in this work is based on a Lagrange–Galerkin finite element method as it offers the best choice for a moving boundary complex flow problem.

**MODELLING STRATEGY**

In order to take into account the constant variations of the flow channel geometry in a tidal waterway a Lagrangian approach in which governing model equations are solved along fluid particle trajectories is used. This amounts to an automatic re-meshing of the solution domain to incorporate variations of its boundaries. Details of the derivation of fluid particle trajectories have been published elsewhere and will not be given here (Petera and Nasseri, 1996). The kinematic equation of trajectory lines are derived as

$$\mathbf{x} - \mathbf{X} = \frac{\Delta t}{2} (\delta + \frac{1}{2} \Delta t_v \nabla \cdot \mathbf{V}_n) \left[ \mathbf{V}_n + \frac{1}{2} \Delta t_{n+1} (\mathbf{V}_n - \mathbf{V}_{n-1}) \right]$$  \hspace{1cm} (11)

where $\mathbf{x}$ is the position vector of a node at current time, $\mathbf{X}$ is the position of feet of the trajectory passing through that node at the previous time level, $\delta$ is Kronecker delta, $\mathbf{V}_n$ and $\mathbf{V}_{n-1}$ are current and previous time level velocity fields, respectively, and, $\Delta t_n$ and $\Delta t_{n+1}$ are time intervals. Equation (11) provides a relationship between present locations of material nodes (which coincide with the nodes in a current mesh) and their previous locations (feet of material particles). In this way, the trajectories that bring material particles (which are field unknowns in our calculations) are determined. Therefore constructing a Lagrangian framework along the particle trajectories described by equation (11) the following solution algorithm can be used to solve the model equations.

**Step 1:** A computational domain, which represents the maximum extent of the flow channel, is selected and discretised into a fixed mesh. The part of the domain that is completely filled with water is called the ‘current mesh’. The nodes within the current mesh are regarded as ‘active nodes’.

**Step 2:** The next stage involves imposition of appropriate initial conditions for the prime unknowns at all nodes. It is known that tidal flow simulations can start with arbitrary initial conditions, i.e., cold start. In the present case, the simulations are carried out with initial values of discharge equal to zero, depths equal to the elevation of the bed at each node and salt concentration equal to 25 ppt.

**Step 3:** Next, the feet of fluid particle trajectories passing through each nodal point in the computational mesh at the current time level are found using equation (11). The coordinates of the feet of fluid particle trajectories $(\mathbf{X}_i)$ at old time step are stored in an array as defined by $\mathbf{X}_i^n$, $i=1,2,\ldots,P$. Here ‘i’ and ‘n’ are the node and element numbers, respectively.

**Step 4:** For every active node in the current mesh, the corresponding location array is searched to find elements inside which the feet of the trajectory passing through the selected node are located. This search is based on the following isoparametric mapping relationships that relate global coordinates of a node to its local coordinates:

$$\mathbf{X}_i = \sum_i N_i(\zeta, \eta) \mathbf{x}_i \quad i=1,2,\ldots,P$$  \hspace{1cm} (12)

The unknowns in equation (12) are local coordinates of the foot, i.e., $\zeta, \eta$. $N_i$ are the shape functions so that for a bi-quadratic element, we have

$$N_i(\zeta, \eta) = \frac{1}{4} \zeta(1-\zeta)\eta(1-\eta)$$  \hspace{1cm} (13)

etc. The justification for using equations (13) is that in the finite element solutions, representing a complex geometry moving domain, are formulated with respect to a master square master element. Within the master element the local coordinate system is always a natural Cartesian system. Therefore the origin is at the centre of the square and its corners have the coordinates of (1,1), (1,-1), (-1,1) and (-1,-1). After insertion of the global coordinates of the feet in the left hand side of equation and the global coordinates of the nodal points in a given element in the right hand side of equations (13), they are solved using the Newton-Raphson method. If the selected foot was actually inside a chosen element, then for a quadrilateral element, its local coordinates must lie between +1 and −1.

Normally, the search for the feet of the fluid trajectories ending on nodes at a current time step requires finding unique and acceptable solutions within the element immediately upstream of the current nodes. In a non-branching domain this presents little difficulty (Nasseri and Kafai, 1999). But at a branch, the non-linear equations (13) may converge to an apparently correct solution but yield physically unacceptable results. This is because it may be possible to find more than one starting point for a current node each located at a different branch. To identify the physically correct result from a converged solution, which is not acceptable, a mass balance based on the computed values at the end of each time step for the junction area is carried out (Das and Nasseri, 2004).
Step 5: The field variables at the foot of each trajectory are calculated at the old time step through finite element interpolation of the old step values. This is done at nodes belonging to those elements where each foot is found to be located.

Step 6: Using the simulated results from step (5), previous time step salt concentration at the feet of fluid particle trajectories and the corresponding values of solute mass are calculated via the solution of the modal equation representing salt balance.

Step 7: Time is incremented and all the steps starting from step (3) are repeated until the end of a tidal cycle.

STRUCTURE OF THE IT TOOL

To achieve the goals of flexibility, practicality, speed, multi-user capability and computing economy the use of the described model is based on its implementation via an Information Processing Tool. The details of this system was presented in a international conference on Industrial Simulation (N.Keshavarzi-Roonizi et al, 2004). The following is a brief description of the system. The IT tool used here is constructed as a combined system consisting of three distinct modules.

These modules can be briefly defined as:

- Front-end, provides a network node for communication between the users and software assets that are in the processor (data generator) and back-end modules.
- Data-generator, is a systematic library of various modelling software that use deterministic or stochastic techniques to simulate, evaluate and generate quantitative results for hydro-environmental problems for which the user seek solution.
- Back-end, provides a network node for the import and processing of the data (i.e. results, answers, evaluations) generated by the software in the generator module. The processed results are logged and returned to a window in the front-end for the utilization by the users.

COMPUTER SIMULATIONS

In order to verify the applicability of the developed IT tool for the complex moving boundary modal finite element scheme we have investigated modelling of salt intrusion in the upper Milford Haven Estuary (Wales, UK). Detailed description of the physical environment of this estuary and its overall hydrodynamic behaviour has been published extensively in other papers and will not be repeated here (e.g. see Das and Nasseri, 2004). The most dominant geometrical feature of this estuary is that it is formed as Eastern and Western Cleddau rivers join at Picton Point. Beyond this confluence the estuary mainly consists of a single dominant flow channel, DauCleddau, Figure 2.

![Figure 2. Upper Milford Haven Estuary](image)

The input data used in the present study are the bathymetry (providing information regarding flow channel depth) of the estuary channel, recorded water surface elevations at the estuary mouth, fresh water inflows at the tidal limits of Eastern and Western Cleddau rivers, open sea salinity of 34ppt at the estuary mouth and zero salinity at its tidal limits. Detailed discussions regarding the choice of these data are published previously and will not be repeated here (Das and Nasseri, 2004).

The main focus of the present modelling has been the simulation of salinity variations at the confluence of the Cleddau rivers. Obviously the momentum distribution and mixing at this location are complex and the ability of the present approach to provide detailed quantitative information about salinity within this junction should be tested. In order to maintain the consistency of the output we have focussed on the inner part of the flow channel at the junction. This is a triangular section with its vertex located at Picton point and its base extends to the limits of deep channel section which remains wet (i.e. water filled) throughout the tidal cycle. The deep inner channel of the estuary is represented by the darker section in Figure 2. As the main purpose of the present work is to obtain results at different depth of the flow channel, focussing on the inner channel we can show a consistent domain at various depth fractions. The extend of the boundary of the inundated channel changes continuously during a tidal cycle and hence at different depth the shape and surface area of the flow channel section is different. However, focussing only on the inner part of the flow channel a consistent section both during flood and ebb can be studied with clarity.

Figure 3 shows simulated salinity contours at the surface of the junction area for incoming tide, two hours before high water at Picton Point. This result clearly shows the
variation of salinity within the surface water at the junction area. In figure 4 salinity distribution corresponding to the same time but at middle of the water column at the junction is shown. As the comparison of figures 3 and 4 shows there is a significant variation in salinity in the vertical direction. Therefore models that rely on depth averaged governing equations which imply perfect vertical mixing do not correspond to reality. In figures 5 and 6 similar data for an ebb tide are shown. Greater mixing of fresh and salt water has reduced the overall salinity but the surface variations and, more significantly, variations between the surface and mid-column salinities are still predicted.

Figure 3. Simulated Salinity 2 hrs before high water at Picton Point, water surface

Figure 4. Simulated Salinity 2 hrs before high water at Picton Point, mid-column surface

Figure 5. Simulated Salinity 2 hrs after the start of ebb at Picton Point, water surface

Figure 6. Simulated Salinity 2 hrs after the start of ebb at Picton Point, mid-column surface

CONCLUSIONS

We have developed a novel modelling technique based on the use of spectral expansions to simulate salinity distribution in estuaries. Due to sophisticated nature of this approach its efficient implementation can be achieved via the use of an IT tool which using interactive front and back ends guides the user to obtain results with ease. The model has been applied to study salt intrusion in Upper Milford Haver Estuary. Preliminary results shown in this paper shows the applicability of the approach to realistic cases. Quantitative results show that the spectral technique has the flexibility to provide detailed information in all three spatial directions under natural conditions.

REFERENCES


ACKNOWLEDGMENT

The authors wish to thank Engineering and Physical Research Council of UK (EPSRC) and Loughborough University for PhD studentships which enabled the development of the Information System described in this work.
TEMPERATURE AND AIRFLOW SIMULATION
INTELLIGENT SYSTEM OF INDIRECT TEMPERATURE MEASUREMENT

Karol Kostúr and Ján Kačur
Institute of Production Process Control, Technical University of Košice
B. Němcovej 3, 043 54 Košice,
Slovakia
E-mail: karol.kostur@tuco.sk

KEYWORDS
Numerical methods, Simulation, Verification, Partial differential equations, Model testing.

ABSTRACT

The aim of the solution is the creation of a system for indirect measurement that can, based on measured temperatures of furnace atmosphere or surface temperatures, determine the temperatures inside the body. The system intelligence is given by two approaches: application of neural networks and the knowledge of the heat transfer processes. The gist of the paper consists in the second approach which is more universal as it enables to take into account the parameters of the load being heated/cooled. The experimental verification was realized on an inhomogeneous load (annealing roll) whose thermophysical parameters are not known. Therefore, a method was developed for their determination which is based on the solution of an optimization task where the aim is to minimize the error between internal measured temperatures and simulated temperatures at verification stage. The difference between the direct checking measurement and indirect measurement of temperatures in both approaches is about 3 %.

INTRODUCTION

At present in the engineering practice there are no measurement systems able to measure the temperature inside heat-treated massive products in a non-destructive way. But the principle is well-known – the indirect measurement. For the measurement of the surface temperature there exist a wide scale of sensors. However, for the measurement of the inner temperature, at present time there is no sensor that, through a non-destructive position inside the product during heating process, could measure inside temperatures. This paper describes some theoretical resources, used within the framework of the solution of the project “Intelligent System of Indirect Temperature Measurement”. In Figure 1 is schematically shown a connection of the system indirect measurement in box furnace. In this case the batch is fixed and is heated from all sides and it is able to provide direct temperature measurement of the material surface (Tmax) or the temperature of the atmosphere in the furnace.

Figure 1: The Diagram of Box Furnace

Available would be a thermocouple or an infrared pyrometer. The unknown temperature inside of the product is then continuously calculated by software from the directly measured temperature.

CLASSIFICATION OF SYSTEMS WITH INDIRECT MEASUREMENT

At present there are many systems of indirect measurement, mutually differing in structure and the number of unknown quantities. The authors (Kostúr et al. 2005a; Bulínová and Kostúr 2003) attempted to create an overview of these systems of indirect measurement. The indirect measurement can be classified by various aspects. Basically, by current experience and knowledge from research of indirect measurement we can make the following classification:

- by the type of the transformation,
- by the complexity,
- by the access of solution,
- by the aspect of the realization.

The Classification of Indirect Measurement by the Type of Transformation

The problem of indirect measurement and a development of indirect measurement methods is still under development and still a subject of research. One of the possible classifications is the classification by the type of transformation. According to this classification we can divide indirect measurement into the following three groups:

a.) the first type is transformation \( x(\tau) \Rightarrow y(\tau) \),

b.) the second type is transformation \( x(\tau) \Rightarrow y(\tau) \),

c.) the third type is transformation \( x(\tau) \Rightarrow y(\tau) \).

where:
\( \mathbf{x} \) is a vector of input values from indirect measurement,
\( \mathbf{y} \) is a vector of indirect measured values.

Let us to emphasize that in the terminology of the control systems we usually use \( \mathbf{y} \) as the vector of output (controlled) values of the process. However, in indirect measurement, according to previous types of transformations we will understand the vector \( \mathbf{x} \) as all the values of input to the indirect measurement, and accordingly also output values from the process that are depicted in Figure 2.

![Figure 2: Input and Output Values in the Indirect Measurement](image)

The designed system for the indirect temperature measurement derives from the directly measurable values, which can be measured normally with technologies, and based on these measurable data, the system continually provides information about inner temperatures for the user, which is not possible to obtain without non-destructive measurement.

The intelligence of the designed system for the indirect temperature measurement can be based on various development methods:

- a) the theory of non-stationary heat conduction, heat transfer by radiation, and convection,
- b) the theory of neural networks,
- c) the theory of the Kalman filter as a predictor,
- d) the stochastic approach.

So far the research (Kostúr et al. 2005b, Kostúr et al. 2005c) has shown unsuitability of application of the Kalman filter. An application of neural networks (Bulnová and Kostúr 2003) appears to be suitable for this case as in this case the error is approximately the same, from the viewpoint of simulated and checked temperatures, as in the case a). The stochastic approach was used in temperatures measurement of the melt in the steel converter (Kostúr et al. 2004), where the error of indirect measurement reached about 3%.

In the following chapter is described the mathematical model, which was used for the assurance of system intelligence, based on the theory of non-stationary heat conduction, heat transfer by radiation and convection. Software will be presented, which applies the mathematical model in the form of a computer algorithm.

**MATHEMATICAL MODEL**

In the framework of the project “Intelligent System of Indirect Temperature Measurement” a need arose to create a mathematical model for non-stationary heat conduction in two-dimensional temperature field. As a starting point, Fourier partial differential equation of heat conduction (1) (Kostúr et al. 2005c) was used.

\[
\frac{\partial t}{\partial \tau} = a \left( \nabla^2 t + \frac{i_0}{\lambda} \right)
\]  

(1)

where:

\[ a = \frac{\lambda}{c \cdot \rho} \]

is the temperature conductivity, that characterises the speed of temperature changes in the material \([m^2 \cdot s^{-1}]\), \( \lambda \) is the coefficient of thermal conductivity, dependent on the type of material \([W \cdot m^{-1} \cdot K^{-1}]\), \( c \) is the specific thermal capacity \([J \cdot kg^{-1} \cdot K^{-1}]\), \( \rho \) is the material density \([kg \cdot m^{-3}]\), \( t \) is the temperature \([K]\), \( \tau \) is the time \([s]\), \( i_0 \) is the internal thermal source \([W \cdot m^{-3}]\).

Equation (1) is a partial differential equation of the second order with right side. Its solution is complicated for \( i_0 = 0 \) even for simple geometric objects. We can solve this equation by using the numerical methods, that consist in dividing the object into the required number of valid chosen plane or space partitions (elements).

**Finite Difference Method**

In this chapter is presented an explicit application of finite difference method for the following problem. It is necessary to solve the mathematical model of heat conduction in two-dimensional objects (Kostúr et al. 2005b):

\[
\frac{\partial t}{\partial \tau} = a \left( \frac{\partial^2 t}{\partial x^2} + \frac{\partial^2 t}{\partial y^2} \right)
\]  

(2)

The steel roll is considered an object.

For boundary conditions of the first type:

\[
T_k = f(x, y, \tau)
\]  

(3)

Initial conditions:

\[
T_p(x, y) = f(x, y, 0)
\]  

(4)

where: \( T_k \) is the temperature of object surface \( T_p \) is object temperature at time \( \tau = 0 \)

We are solving the mathematical model (2) by the net method in coordinate system \( x, y \). On the object (in cross-
section of the roll) we create the net by drawing parallel lines with the coordinate axis at distance \( \Delta x, \Delta y \) (Figure 3).

\[ T_{i,j,k+1} = \left( 1 - \frac{2\alpha \cdot \Delta t}{\Delta x^2} - \frac{2\alpha \cdot \Delta t}{\Delta y^2} \right) T_{i,j,k} + \frac{\alpha \cdot \Delta t}{\Delta x^2} (T_{i+1,j,k} + T_{i-1,j,k}) + \frac{\alpha \cdot \Delta t}{\Delta y^2} (T_{i,j+1,k} + T_{i,j-1,k}) \]  

(5)

On the ground of equation (5) we can calculate the temperature at optional node point in the next time step in the case when we know the temperature at node points, where the point is included in the previous time step. Further analysis of equation (5) shows that the first member in parentheses causes divergence in calculations if the member assumes negative values.

**STRUCTURE OF SIMULATION MODEL.**

The heart of the simulation model is program algorithm which solves equation (5) from previous chapter. The program calculates the temperature at all points of internal matrix shown in Figure 3. The temperatures at boundary points of the matrix are regarded as known in every time step of the simulation. These temperatures were measured during experimental heating of the roll in the furnace or in online mode, and were obtained continually during the measurement. In every subsequent time step (step \( k+1 \)) the temperatures at the points of internal matrix are calculated from known temperatures of the matrix from previous time step (step \( k \)). This solution is well-known as the solution with boundary conditions of the first type. In Figure 3 is also shown the location of thermocouples, projected into the space of temperature points field (matrix). In Figure 4 is shown a block diagram of the simulation model structure. The computer model was created according to this diagram.

![Block Diagram of Simulation Model Structure](image)

![The Main Window of the Program for Modeling of Temperature Points Field](image)
in °C or in K. and the user can display the time behaviour of temperatures of every point from the point matrix.

**VERIFICATION OF MODEL**

From experimental measurements on the laboratory roll a few simulations have been made (Table 1). Also we have two measurements on real steel rolls from annealing plant (Table 2). With the data from the annealing plant real dimensions of the roll were used and another division of point matrix. The degree of correspondence between the behaviour of the measured temperature and simulated temperature from the model was quantitatively expressed by the relative error \( \delta \) according to equation (6) and with maximum difference of the temperature during measurement (Kostúr et al. 2005b).

\[
\delta = \frac{\sum_{k=1}^{n} |T_{k}^{\text{mod}} - T_{k}^{\text{meas}}|}{n} \cdot 100 \quad [\%]
\]  

where:

- \( T_{k}^{\text{mod}} \) is the temperature calculated by model [°C]
- \( T_{k}^{\text{meas}} \) is the measured temperature [°C]
- \( n \) is the total count of time steps from simulation
- \( k \) is the time step \((k=1, \ldots, n)\)

We made an analysis of results from the model and measurements in MS Excel. Results of the analysis are presented in the following tables:

**Table 1: Simulations with Laboratory Roll**

<table>
<thead>
<tr>
<th>Measurement from 13. 10. 2005:</th>
<th>T2</th>
<th>T3</th>
<th>T4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Comparative temperature</td>
<td>35.38</td>
<td>59.85</td>
<td>30.95</td>
</tr>
<tr>
<td>Relative error [%]</td>
<td>4.40</td>
<td>6.83</td>
<td>4.33</td>
</tr>
<tr>
<td>Max. difference [°C]</td>
<td>23.50</td>
<td>27.48</td>
<td>13.80</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Measurement from 23. 11. 2005:</th>
<th>T2</th>
<th>T3</th>
<th>T4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Relative error [%]</td>
<td>5.03</td>
<td>3.86</td>
<td>2.31</td>
</tr>
<tr>
<td>Max. difference [°C]</td>
<td>51.46</td>
<td>26.99</td>
<td>13.80</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Measurement from 25. 11. 2005:</th>
<th>T2</th>
<th>T3</th>
<th>T4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Relative error [%]</td>
<td>7.73</td>
<td>4.51</td>
<td>1.80</td>
</tr>
<tr>
<td>Max. difference [°C]</td>
<td>61.12</td>
<td>29.57</td>
<td>15.40</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Measurement from 28. 11. 2005:</th>
<th>T2</th>
<th>T3</th>
<th>T4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Relative error [%]</td>
<td>48.97</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

\( T_2, T_3, T_4 \) is labeling for the thermocouples for points, in which the temperature from measurement with temperature from the model was compared. The labeling of thermocouples is according to the diagram in Figure 3.

**Table 2: Simulations with Real Rolls from Annealing Plant**

<table>
<thead>
<tr>
<th>Measurement from 1. 3. 2005 Roll D:</th>
<th>ISA</th>
</tr>
</thead>
<tbody>
<tr>
<td>Comparative temperature</td>
<td>5.03 %</td>
</tr>
<tr>
<td>Relative error [%]</td>
<td>47.07</td>
</tr>
<tr>
<td>Maximum difference [°C]</td>
<td>2.19</td>
</tr>
<tr>
<td>Maximum difference [°C]</td>
<td>23.80</td>
</tr>
</tbody>
</table>

LabeIs ISD, ISA, and ISC correspond to labels of thermocouples for the measurements from annealing plant. These thermocouples are placed inside rolls marked as roll D, roll A, and roll C. The position of thermocouples is shown in Figure 6. The corresponding thermocouples were engaged as comparative and analogously to thermocouples \( T_3 \) (on laboratory roll) were placed inside the roll. Thermocouples marked as HSD, HSC, CSD, HSB, CSC, CSB, HSA, CSA in Figure 6, were used for direct measurement of surface temperatures. In the following figures are graphic comparisons of results from simulations and measurements.

**Figure 6: Position of Thermocouples on Four Rolls in Annealing Plant**

**Figure 7: Comparison of Direct and Indirect Measured Temperature from 11. 7. 2005, Roll C, Annealing Plant**

In Figure 7 is shown the time behaviour of measured temperature inside the roll and simulated temperature,
which was simulated from measured boundary temperatures. The measurement was performed on a real steel roll from annealing plant. The figure demonstrates good equality between the measurement and simulation. Compared was the temperature 15°C inside the roll, that was directly measured by the thermocouple. The correspondence is also quantitatively expressed by the relative error. Its value in this case was only 2.19% and maximum difference was 23.80 °C that was found only during the heating (at time 600 min). In Figure 8 is shown a visual comparison of equality between measurement and simulation, which were performed on laboratory roll. The roll was heated in the laboratory furnace. The technical realization of the measurement, was based on programmable logical controller (PLC) and computer. In Figure 8 is shown the behaviour of the measured and simulated temperatures, or calculated temperature. The relative error found in this case reached 3.20% and the maximum difference was 27.48 °C, also found in half-time of heating. Compared was the temperature measured by thermocouple $T_3$, whose location inside the roll is demonstrated by the diagram in Figure 3.

![Diagram showing measurements and calculations](image_url)

**Figure 8: Comparison of Direct and Indirect Measured Temperature from 23. 11. 2005, Laboratory Roll**

It has been found that in these cases, from the viewpoint of measurement error, a detailed measurement of the surface (pyrometer) is not needed and it is sufficient to perform measurements at representative points $T_{18}$, $T_1$, $T_6$, $T_{15}$, $T_7$, $T_{17}$, $T_5$, $T_6$, $T_{12}$, $T_10$. see Figure 3. For individual surface points the temperature was determined with linear approximation. Another finding is the transformation of heat transfer in porous bodies through radiation, convection (air spaces) into heat conduction through heat conductivity $\alpha$. The simulated temperatures with the aid of so-called corrected heat conductivity agree better with measured heat temperatures. The relative error by using the corrected heat conduction decreased from 3.2% to 3.19% and the maximum deviation was 25.9 °C. The above comparison was made on the basis of measured and simulated temperature of thermocouple $T_3$.

**CONCLUSION**

This paper basically presents the results from research of the project “Intelligent system of Indirect Measurement of Temperature”. The existence of such intelligent measurement system enables to increase the quality of products and to decrease energy consumption in energetically costly technologies, as are no doubt technologies of heating in industrial furnaces. In this paper was presented software that represents the intelligence of the designed system of indirect measurement. The complex system of indirect temperature measurement contains algorithm for correction of thermo-physical parameters $a$, $\lambda$, $c$. These algorithms are in details described in current report on the project and another published papers related to the project. The essence of these algorithms is solving optimisation task:

$$\text{to minimize } \sum_{i=1}^{p} \delta_i$$

where $p$ is the number direct measured controlling temperatures during verification. The software presented was verified in laboratory conditions on the model of steel roll and on measurements from annealing plant.

The proposed work proves that it is possible to indirectly measure temperatures inside a body with sufficient engineering accuracy. The originality of the research is in the use of so-called corrected heat conductivity by means of which it is possible to eliminate the influence of other parameters and processes such as e.g. inhomogeneity of material or heat transfer in a way different than conduction.

**ACKNOWLEDGEMENT**

This work was supported by project of applied research with number 4/0016/05 from the Ministry of Education of Slovak Republic.

**REFERENCES**


AERODYNAMIC AND HEAT TRANSFER CHARACTERISTICS
OF TWO INTERFERING LOW-RISE BUILDINGS

Osama E. Abdellatif
Shoubra Faculty of Eng., Benha Univ., Egypt.
Email: sameh_sohasmr@yahoo.com

ABSTRACT
This study represents an experimental and computational investigation of the aerodynamic and
heat transfer characteristics of two interfering low-rise
buildings. The two buildings situated in line are
similar with a slanted roof. The investigated
parameters include the gap between the two buildings,
the roof angle, and the wind speed (Reynolds number).
The solar radiation absorbed by the roof
and conducted into the building that appears as a cooling
load is also studied. Heat flow through such roofs is
found to be sensitive to number of factors, including,
wind speed, roof angle and surface to air temperature
difference.
The experimental results demonstrated pressure
distributions on the different walls and roofs of the
two buildings. Also, the drag force on the two
buildings is measured for different interfering cases.
The corresponding numerical results were obtained
using the computational finite element procedure
(ANSYS 9.0 for two-dimensional model and Fluent
6.12 for three dimensional model) that uses the
standard k-ε model. Computational results include
velocity vectors, distributions of turbulence kinetic
energy, and temperature gradients as well as the
pressure distribution. Certain consideration was paid
to the difference in aerodynamic characteristics
between a single building and two interfering
buildings. Comparison between the numerical and
experimental results showed a good agreement in
terms of gross feature of mean flow for all cases
examined, although some detailed differences were
observed.

INTRODUCTION
Low-rise buildings are defined as structures with a
ratio of mean roof height to least horizontal dimension
less than 1 and the mean height less than 18 meter.
Wind loading on low-rise building is primarily
dependent on the highly turbulent flow patterns
generating the building. The flow patterns around
the building envelope induced due to wind turbulence
and building generated turbulence would result in wind
loading on low-rise structures. These wind loads are
varying in nature due to spatial and temporal variation
of turbulence in the atmospheric boundary layer in
which these structures are submerged. They are
expressed as the non-dimensional pressure coefficients
which are dependent on the site-specific wind and
terrain characteristics and building geometry.
The flow over a typical low-rise building can be
classified as bluff bodies with respect to the flow
around them, in contrast to streamlined flows such as
occurs around aircraft wings. In streamlined flows the
flow streamlines closely follow the shape of the body,
in contrast to flow around bluff bodies where the flow
separates at the leading edge corners. The separated
flow region is divided from the outer flow by a thin
region of high shear called the free shear layer.
Similar flow separation and re-attachment occurs on
the side walls of low-rise buildings, although the
magnitude of the mean pressure coefficients tends to
be lower. Mean pressures on windward walls tend to
be positive, while leeward walls generally experience
negative pressures.
However, the aerodynamic field around a building is
expected to be affected by many parameters. Such
parameters may include the shape of the building,
Reynolds number, and the roof angle. The
characteristics of the aerodynamic field affect the
overall drag on the building, suction force on the roof,
internal ventilation, comfort of the residents, thermal
field inside the building, etc. The situation becomes
even more complicated when two buildings are
interfering. Building interference means that the two
buildings are such close to each other that they have
mutual effect on the aerodynamic fields of each other.
In the past four decades, several full-scale field
measurement programs have been undertaken to study
the wind effects on low-rise structures. As reported by
Kumar (2005), the first study was carried out on a
two-story house built in early 1970’s in Aylesbury,
England. It had dimensions of 13m x 7 m as the length
and the breadth, and the eaves height was 5 m. The
building roof pitch varied, ranging from 5 to 45°
Comparative wind tunnel studies on a 1:100 scale
model of this building were carried. Based on the
results reported from these wind tunnel studies, it was
found that the lab-to-lab variation in pressure
coefficient was attributed to difference in the method
of data acquisition and in the measuring point of the
reference static pressure and dynamic pressures.
Another gabled roof, single story building was constructed in Silsoe, England in late 1980s. In the plan form, dimension of the building are represented as 24.13m x 12.95m as the length and the breadth while the eave height was 4 m and the roof pitch angle was 10 degree. Wind-tunnel test on 1:100 scale model of the building have been done and a close agreement between model and the full-scale pressure coefficient data were obtained. A unique full-scale test facility was established in Lubbock, Texas in the late eighties for measuring wind pressures and their effects on the test facility structure. This facility, popularly known as the Wind Engineering Research Field laboratory (WERFL), is a typical low-rise building. The facility is unique in its ability to provide a combination of various measuring capabilities not available in the previous full-scale facilities. In its plan form, the building has dimensions of B x L x h =9.1 m x 13.7 m x 4.0 m, with a nearly flat roof, unlike the other previously tested full-scale facilities. The field site experiences sustained winds of 9-16 m/s throughout the year. A number of wind tunnel studies were conducted on different scale models of WERFL to simulate the pressure coefficients under controlled conditions, Sury (1992).

Actually, building-interaction was investigated by many researchers. Baskaran and Stathopoulos (1994) reviewed the state-of-the art in computational wind engineering, including the finite element, finite difference and control volume methods for wind engineering problems. Their computations have been made to variety of building configurations, including computation of normal wind flow conditions for a building with different aspect ratios, and modeling wind environmental conditions around a group of buildings. They stated that the computer modeling technique may eventually enhance the design of buildings and structures against wind loading and supplement the current design practice of using building codes and standards or performing experiments in wind tunnels. Vlachogiannis et al. (2003) carried out a thorough numerical modeling investigation of the wind field and dispersion of pollutants within the space between slant-roof buildings for neutral stability conditions corroborating the calculated results with experimental measurements from a wind tunnel. The comparisons of the model results with the experimental pollutant concentration data yielded a good agreement and any discrepancies found were within a factor of two. The factor of two on concentration predictions is well within the expected uncertainties derived from such modeling where the spread of the model results relative to wind tunnel data was found to be between a factor of 5 and 100. The simulated wind field compared well to the wind tunnel field, with a vortex formed between the buildings, aiding the space ventilation from the pollutants emitted from the source. The different release cases showed that the wind fields and concentrations were largely independent of the direction of the source jets. Miles and Westbury (2003) compared surface pressures and pedestrian-level wind speeds obtained from CFD and wind tunnel testing for a group of buildings immersed within an urban boundary layer. Their results demonstrated an important sensitivity of the CFD predictions to the way in which the urban boundary layer is represented in the upstream region. They suggest that further work is needed to investigate the most appropriate way to incorporate such boundary layers. One option, which was shown in their study to have some benefit, is to include roughness blocks explicitly within the CFD geometry in a manner similar to that used in wind tunnels. Geurts et al. (2004) carried out wind tunnel experiments to provide data to determine design loads for the elements in facades and roofs. They stated that no design rules are available for the influence of nearby buildings on the local loads in densely populated areas. Thus, wind tunnel measurements are the only reliable technique available. Their work gave recommendations on where to focus future research with respect to the reliability of the local wind loads.

Analysis of the thermal field around and inside buildings was also carried out by many researchers. Charvat et al. (2001) dealt with the CFD modeling of airflow and temperature fields of a glazed attic, and comparison of results with experimentally obtained data. Their main goal was to find optimal locations of ventilation system inlets in order to increase the efficiency of solar energy utilization. Their results implied that the inlets of the ventilation system should be located just underneath the roof. Choudhary and Malkawi (2001) combined physical and computational simulations within a unique framework for the aim of establishing a methodology for micro-level building thermal analysis. They stated that the most limiting factor of the combined methodology is that real-building scenarios often include complex geometries. The building geometry would have to be abstracted and simplified in order to model it with the CFD prototype. Hien et al. (2004) examined the impact of façade material properties on indoor environment and façade thermal performance. Their investigation included field measurements, laboratory test, and computer simulations. Their results of field measurement conducted in 4 school buildings and 2 HDB buildings revealed that the current façade materials which were used in these buildings have a good thermal protection except the aluminum panel. Clear et al. (2001) measured the surface heat transfer on the roof of two commercial building and developed a correlation expresses the outside convective air-film coefficient for flat roofs as a function of surface to air temperature difference, wind speed, wind direction, roof size, and surface roughness. Their correlation was accurate for horizontal roof and for roof tiles up to
about 20° Many aspects about solar heating of buildings can be found in (11).

The objective of the present study is to investigate experimentally and numerically the aerodynamic characteristics of two interfering low-rise similar buildings situated in line with slanted roof. The investigated parameters include the gap between the two buildings, the roof angle and the wind speed. Also, the effect of solar radiation absorbed by the roof and conducted into the building that appears as a cooling load is also studied. Heat flow through such roofs is found to be sensitive to the above mentioned parameters and surface to air temperature difference.

EXPERIMENTAL SETUP

All of the aerodynamic experiments were performed in an open circuit, atmospheric exhaust tunnel. Room air is pushed through a settling chamber with honeycomb-type flow straighteners and a 3.33:1 contraction into a test section of 0.3m x 0.3m. The flow is created by a two-inlet centrifugal fan that is driven by a 4 hp AC motor. The free-stream velocity was adjusted at two values, typically: 15 and 25 m/s. These values of velocity correspond to Re = 81000 and 135000, respectively. Three sets of models were constructed to measure the static pressure distributions on the external walls of the roof buildings. Figure 1 shows a general sketch of the test section of the wind tunnel and models. Each set consists of two similar models with exactly the same roof angle. Three roof angles were set, namely: 30°, 40°, and 50°. Each model was fabricated from 4 mm-Plexiglas sheets. The models have square base cross-section (0.10m x 0.10m). The models maximum height, $H_{max}$, equals to 0.10 m. The models were equipped with pressure taps that cover the surfaces of the walls and roof of the models. The pressure taps were connected to a multi-manometer through rubber tubes. Locations of concentration measurement for slanted roof with $\Delta = 30$° are shown in figure 2.

NUMERICAL INVESTIGATION

Flow and heat transfer over low-rise buildings were also simulated with two types of computational fluid dynamics software. Incompressible, 2D, turbulent flow around the buildings was first calculated using the computational finite-element procedure ANSYS (Version 9.0) using standard k-H model. Over-predicted values of the pressure coefficient were noticed. Three dimensions model using FLUENT (Version 6.12) was used. This software is based on a finite volume discretization of the equations of motion, an unstructured grid volume made of either rectangular prisms or tetrahedral cells. General views of the computational mesh for both models are shown in figure 3.
Governing Equations and K- H Model

The governing equations for the mean velocity and pressure are the Reynolds averaged Navier-Stokes equations for incompressible flow. The governing equations of flow and thermal fields as well as K-H model can be written as:

- Mass:
  \[ \frac{\partial U_j}{\partial x_j} - \frac{1}{Y} \frac{\partial P}{\partial Y} - \frac{\partial}{\partial x_j} \left( \frac{\partial U_j}{\partial x_j} \right) = 0 \]  
\[ \text{(1)} \]

- Momentum:
  \[ \frac{\partial}{\partial x_j} \left( \frac{U_j}{Y} \right) = \frac{2}{3} \frac{K}{Y} \frac{\partial P}{\partial Y} - \frac{2}{3} K \frac{\partial Y}{\partial Y} \frac{\partial U_j}{\partial x_j} \]  
\[ \text{(2)} \]

- Turbulent viscosity:
  \[ \frac{\partial \Theta}{\partial x_j} \left( \frac{C_{\mu} K^2}{H} \right) \frac{1}{2} \left( \frac{\partial U_j}{\partial x_j} + \frac{\partial U_j}{\partial x_i} \right) \]  
\[ \text{(3)} \]

- Energy:
  \[ \frac{\partial}{\partial x_j} \left( \frac{U_j}{K} \right) = \frac{\partial}{\partial x_j} \left( \frac{\partial \Theta}{\partial x_i} \right) \]  
\[ \text{(4)} \]

- Turbulence kinetic energy:
  \[ \frac{\partial}{\partial x_j} \left( \frac{C_{\gamma} K^2}{H} \right) \frac{1}{2} \left( \frac{\partial U_j}{\partial x_j} + \frac{\partial U_i}{\partial x_i} \right) \]  
\[ \text{(5)} \]

- Dissipation of turbulence kinetic energy:
  \[ \frac{\partial}{\partial x_j} \left( \frac{U_j}{K^2} \right) = \frac{\partial}{\partial x_j} \left( \frac{\partial \Theta}{\partial x_i} \right) \]  
\[ \text{(6)} \]

- Single Building:

Fig. 4 shows the wind tunnel measurements for velocity distribution through the test section at different y/z position. It is clear that most of the building model is subjected to almost uniform flow.

RESULTS AND DISCUSSIONS

Both the experimental and numerical results for the flow and thermal fields are presented and discussed. Experimental results are concentrated on the distributions of pressure coefficient (Cp) on the front and back faces of the front and rear buildings. The other flow characteristics (such as velocity vectors as well as contours of stream-wise velocity, Cp, kinetic energy and temperatures) are found numerically. The integration of both experimental and numerical results gives a complete idea about the flow field around the two interfering buildings.

Computational Aspects and Boundary Conditions

The computations were carried out using quadrilateral elements and the elements are condensed near and between the buildings. For the grid points next to the buildings and near the ground, the law of the wall was applied. For the far field, a symmetry boundary condition was used. The approaching velocity profile is prescribed by a power law. The power law takes the form:

\[ u(y) = \frac{V_s}{y_{ref}} \left( \frac{y}{y_{ref}} \right)^n \]  
\[ \text{(7)} \]

Where u(y) is the velocity at height y. Zero gradients in the flow direction are imposed on the outlet plane. On the solid surface, K = 0 is used and Hs extrapolated from the neighborhood using \( \frac{\partial H}{\partial n} = 0 \).
separation tends to produce negative pressures that act over the entire roof surface. At roof pitches greater than 30° positive mean pressures occur on the upwind roof face, while fully separated flows with flow re-attachment occur downwind of the ridge giving relatively uniform negative mean pressures on the downwind roof slope. This may explain the increase of \(C_D\) as \(\theta\) increases as will be seen later. Figure 6 shows comparison of pressure coefficient along the mid-plane section between the present results and other studies for both experimental and numerical investigation. Due to the lack of available data for pressure coefficient over pitch roof building, the comparison was carried out with the cases of flat roof building that were investigated by others. Generally, the present values of \(C_p\) have the same trend as those of others. The noticed differences on the front face because the other investigation included the effect of wind shear, which is not the case of the present study.

![Fig. 5](image1)

**Fig. 5** Pressure coefficient distribution along the mid-plane for different roof angles. (a) Re=81000 (b) Re=135000

![Fig. 6](image2)

**Fig. 6** Comparison of pressure coefficient along the mid-plane section between the present study and other studies. (a) Experimental. (b) Numerical.

**Two Interfering Buildings:**

Figures 7 and 8 show comparison between the experimental and numerical distribution of \(C_p\) for the two interfering (front and rear) buildings with two values of the gap \((G/W = 0.5\) and \(1.0\)) at \(\Delta = 30^\circ, 40^\circ,\) and \(50^\circ\) for \(Re=135000\). Comparing each set of data from wind tunnel and numerical simulation, most of the results showed that CFD software, Fluent 6.12, can well predict the actual wind tunnel simulation data by choosing optimum boundary condition, grid resolution and turbulence model. The calculated results agree well with the measurements in the wind tunnel by running k-ε turbulence model. The numerical and experimental results for \(C_p\) contours for the two interfering buildings with two values of the gap for \(\Delta = 30^\circ\) are shown in figures 9 and 10.
Fig. 7 Comparison between the experimental and numerical distribution of $C_p$ for the two interfering (front and rear) buildings for $Re=135000$ with the gap ($G/W=0.5$) at (a) $\Delta=30^\circ$, (b) $\Delta=40^\circ$, and (c) $\Delta=50^\circ$.

Fig. 8 Comparison between the experimental and numerical distribution of $C_p$ for the two interfering (front and rear) buildings for $Re=135000$ with the gap ($G/W=1.0$) at (a) $\Delta=30^\circ$, (b) $\Delta=40^\circ$, and (c) $\Delta=50^\circ$. 
Interference of the two buildings has certain main effects, namely: (i) negative values of $C_p$ appear on the front roof of the front building. This means that the front roof is exposed to an unexpected lift force. (ii) Both the front face and roof of the rear building are dominated by negative values of $C_p$. This suggests that the values of $C_p$ of the rear building are very small. (iii) When observing the values of $C_p$ on the back face of the rear building, negative values of $C_p$ may be expected. This means that the direction of the drag force is opposite to wind direction. (iv) Effect of the gap is noticed on the front and back faces of both the front and rear buildings. For $\Delta = 40^\circ$ (Figs. 11 and 12), no signs of negative pressure are noticed on the front roof of the front building. Effect of the gap width ($G/W$) is noticed on the back face and roof of the front building as well as the faces and roofs of the rear building. Figures 13 and 14 show the distributions of $C_p$ for the front and rear buildings, respectively, at $\Delta = 50^\circ$, as expected an increase of the positive values of $C_p$ is noticed on the front roof of the front building.

When considering values of $C_p$, it is clear that values of $C_p$ on the front building increases with $\Delta$. However, small changes of $C_p$ is expected for the rear building with changing $\Delta$. 

Figures 15 shows the numerical results of the distributions of different flow characteristics at $\Delta = 30^\circ$, $Re = 135000$ for two gap values ($G/W = 0.5$ and 1.0).
The following points can be noticed: (i) The effect of gap is noticed at $G/W = 1.0$ for the contours of stream-wise velocity. A zone of high velocity ($U/V = 1.34$) is noticed above the front building. (ii) Small activity is noticed between the two buildings. (iii) Maximum values of the kinetic energy is found at the connection between the front face and front roof of the front building. (iv) Increasing the gap between the two buildings increases the values of velocity inside the gap.

Fig. 15 Numerical results of (a) stream-wise velocity contours, $\Delta = 30^\circ$ (b) velocity vectors, and (c) turbulence kinetic energy (Up: $G/W=0.5$, down $G/W=1.0$).

The flow characteristics of $\Delta = 40^\circ$ for $G/W = 0.5$ and 1.0 are shown in fig. 16. It is noticed that there is an increase in the turbulent kinetic energy above the area between the two buildings especially for $G/W=1.0$.

Fig. 16 Numerical results of (a) stream-wise velocity contours, $\Delta = 40^\circ$ (b) velocity vectors, and (c) turbulence kinetic energy (Up: $G/W=0.5$, down $G/W=1.0$).

Figures 17 show the flow characteristics at $\Delta = 50^\circ$ for $Re=135000$. Maximum stream-wise velocity is found near the tip of the roof. Maximum kinetic energy has moved away from the front building towards the area above the rear building. The rear building is entirely contained in the wake (nearly constant negative pressure) of the front building.

![Flow characteristics](image)

**BUILDINGS DRAG**

The aerodynamic drag of the buildings in different conditions is recorded in the form of the drag coefficient ($C_D$). $C_D$ was calculated by numerical integration of the pressure distributions on the front and back faces and roofs. Table 1 gives the values of $C_D$ for the different tested cases. As can be seen the values of $C_D$ for the single buildings, increases with $\Delta$. These values compare well with the published data of others. For $\Delta=27^\circ$, $C_D = 1.06$ (Oliveira and Younis, 2000) and for $\Delta=40^\circ$, $C_D = 1.37$ (Abdelatif and Abdel-Gwad, 2005). Differences may be attributed to the small variations in the shape of the roofs. For interfering buildings, the interference has a small effect on the front building. Thus, values of $C_D$ of the front building are slightly less than those of the corresponding single building. The interference has a noticeable effect on values of $C_D$ of the rear building. $C_D$ takes very small positive values or even negative values for $(G/W = 0.5)$. This may be attributed to the fact that the rear building is contained completely in the wake of the front building. The effect on the rear building increases with $\Delta$.

![Numerical results](image)

**Table 1 Values of $C_D$ for different cases.**

<table>
<thead>
<tr>
<th>No.</th>
<th>Case</th>
<th>Roof angle ($\Delta$)</th>
<th>Gap ($G/W$)</th>
<th>Drag Coefficient ($C_D$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Single</td>
<td>$30^\circ$</td>
<td>------------</td>
<td>1.153</td>
</tr>
<tr>
<td>2</td>
<td>Single</td>
<td>$40^\circ$</td>
<td>------------</td>
<td>1.312</td>
</tr>
<tr>
<td>3</td>
<td>Single</td>
<td>$50^\circ$</td>
<td>------------</td>
<td>1.44</td>
</tr>
<tr>
<td>4</td>
<td>Front</td>
<td>$30^\circ$</td>
<td>0.5</td>
<td>1.08</td>
</tr>
<tr>
<td>5</td>
<td>Front</td>
<td>$30^\circ$</td>
<td>1.0</td>
<td>1.13</td>
</tr>
<tr>
<td>6</td>
<td>Rear</td>
<td>$30^\circ$</td>
<td>0.5</td>
<td>0.04</td>
</tr>
<tr>
<td>7</td>
<td>Rear</td>
<td>$30^\circ$</td>
<td>1.0</td>
<td>0.23</td>
</tr>
<tr>
<td>8</td>
<td>Front</td>
<td>$40^\circ$</td>
<td>0.5</td>
<td>1.28</td>
</tr>
<tr>
<td>9</td>
<td>Front</td>
<td>$40^\circ$</td>
<td>1.0</td>
<td>1.26</td>
</tr>
<tr>
<td>10</td>
<td>Rear</td>
<td>$40^\circ$</td>
<td>0.5</td>
<td>-0.11</td>
</tr>
<tr>
<td>11</td>
<td>Rear</td>
<td>$40^\circ$</td>
<td>1.0</td>
<td>0.027</td>
</tr>
<tr>
<td>12</td>
<td>Front</td>
<td>$50^\circ$</td>
<td>0.5</td>
<td>1.36</td>
</tr>
<tr>
<td>13</td>
<td>Front</td>
<td>$50^\circ$</td>
<td>1.0</td>
<td>1.35</td>
</tr>
<tr>
<td>14</td>
<td>Rear</td>
<td>$50^\circ$</td>
<td>0.5</td>
<td>-0.23</td>
</tr>
<tr>
<td>15</td>
<td>Rear</td>
<td>$50^\circ$</td>
<td>1.0</td>
<td>-0.067</td>
</tr>
</tbody>
</table>
THERMAL FIELD

The effect of roof angle, Reynolds number, and gap on the heat transfer from the heated roofs of the two buildings is presented in Figures 18-19. The ambient temperature was kept at 20°C (293 K). A constant value of heat flux was applied on the two roofs of each building. Generally, the temperature increases on the roofs of the rear building with increasing roof angle (Δ). Naturally, the temperature on the front roof of the two buildings decreases with increasing Re. An exception is noticed for Δ = 50°, the temperature on the front roof of the rear building is slightly affected by Re. The effect of the gap is mainly noticed on the rear building. Generally, good combination between the roof angle and the gap value may give the desired cooling of the two interfering buildings.

Fig. 18 Distributions of temperature over roofs G/W=0.5 for Δ = 30°, Δ = 40° and Δ = 50°

CONCLUSIONS

From the above discussions, the following points can be stated. The effect of the gap is noticed on the front and back faces of both the front and rear buildings. For Δ = 30°, interference causes the appearance of negative values of C_p on the front roof of the front building. This means that the front roof is exposed to an unexpected lift force. The effect of gap is noticed at G/W = 1.0 for the contours of stream-wise velocity. A zone of high velocity is noticed above the area between the two buildings. Also, increasing the gap between the two buildings increases the values of velocity inside the gap.

As in increase maximum stream-wise velocity is found near the tip of the roof. Maximum kinetic energy has move away from the front building towards the area above the rear building. Also, the rear building is entirely contained in the wake (constant negative pressure) of the front building. Generally, the values of C_p, for the single buildings, increases with Δ. For interfering buildings, the interference has a small effect on the values of C_p of the front building. Thus, values of C_p of the front building are slightly less than those of the corresponding single building. The interference has a noticeable effect on values of C_p of the rear building. C_p takes very small positive values or even negative values for (G/W = 0.5).

The temperature increases on the roofs of the rear building with increasing roof angle (Δ). The effect of the gap is mainly noticed on the rear building. Generally, good combination between the roof angle and the gap value may give the desired cooling of the two interfering buildings.

REFERENCES


NOMENCLATURE

- \( A_p \): projected area of the building.
- \( C_{gh} \), \( C_{sh} \) and \( C_{gh} \) : numerical constants.
- \( C_p \) : drag coefficient.
- \( C_y \) : pressure coefficient.
- \( F_D \) : drag force on the building.
- \( G \) : gap between models.
- \( H_{\infty} \) : model height.
- \( K \) : turbulence kinetic energy.
- \( K_n \) : dimensionless turbulence kinetic energy \((K(V_{sh} V_{sh}))/\).
- \( n \) : normal direction.
- \( Pr_l \) : laminar Prandtl number.
- \( Pr_t \) and \( Pr_{sh} \) : Prandtl numbers for kinetic energy of turbulence and rate of dissipation.
- \( Re \) : Reynolds number.
- \( T \) : mean temperature.
- \( u \) and \( v \) : the stream-wise and cross-wise velocities.
- \( U \) : velocity vector.
- \( U_i \) : velocity component in \( X_i \)-direction.
- \( V_{n} \) : dimensionless velocity component in \( X_i \)-direction \((U/V_n)\).
- \( V_{sh} \) : free-stream velocity.
- \( W \) : width of the building model.
- \( X \) and \( Y \) : the stream-wise and cross-wise directions.
- \( x \) : dimensionless stream-wise distance \((X/W)\).
- \( y \) : dimensionless cross-wise distance \((Y/H_{\infty})\).
- \( Y_{sh} \) : thickness of ground shear layer.
- \( X \) and \( Y \) : the stream-wise and cross-wise directions.

GREEK

- \( \Delta \) : roof angle.
- \( \alpha \) : power law exponent.
- \( \dot{\varepsilon} \) : rate of dissipation of turbulence kinetic energy.
- \( \nu \) : dynamic viscosity.
- \( \nu_{k} \) : kinematic viscosity.
- \( \nu_{t} \) : turbulent kinematic viscosity.
- \( \gamma \) : density.

Abbreviations

- 2-D : two-dimensional.
- CFD : Computational Fluid Dynamics.
- HDB : Housing and Development Board.

Subscripts

- \( M \) : maximum.
- \( \phi \) : free-stream.
AUTHOR LISTING
<table>
<thead>
<tr>
<th>Author</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Abdellatif O.E.</td>
<td>520</td>
</tr>
<tr>
<td>Abdel-Naby S.</td>
<td>49</td>
</tr>
<tr>
<td>Abdulmalek F.</td>
<td>93</td>
</tr>
<tr>
<td>Aghakashi M.</td>
<td>504</td>
</tr>
<tr>
<td>Ahonen-Jonnarh U.</td>
<td>498</td>
</tr>
<tr>
<td>Aiordachioae D.</td>
<td>237</td>
</tr>
<tr>
<td>Alexandre C.</td>
<td>459</td>
</tr>
<tr>
<td>Alfonseca M.</td>
<td>493</td>
</tr>
<tr>
<td>Alitavoli M.</td>
<td>504</td>
</tr>
<tr>
<td>Allet S.</td>
<td>251</td>
</tr>
<tr>
<td>Angeli C.</td>
<td>229</td>
</tr>
<tr>
<td>Arendt M.</td>
<td>442</td>
</tr>
<tr>
<td>Armesto L.</td>
<td>143</td>
</tr>
<tr>
<td>Arnabile M.S.</td>
<td>340</td>
</tr>
<tr>
<td>Ascia G.</td>
<td>285</td>
</tr>
<tr>
<td>Augello A.</td>
<td>86</td>
</tr>
<tr>
<td>Bagheri A.</td>
<td>504</td>
</tr>
<tr>
<td>Bahri P.A.</td>
<td>62/223</td>
</tr>
<tr>
<td>Balas B.</td>
<td>265/269</td>
</tr>
<tr>
<td>Barbosa J.M.A.</td>
<td>182</td>
</tr>
<tr>
<td>Basiri S.</td>
<td>504</td>
</tr>
<tr>
<td>Basto J.A.</td>
<td>415</td>
</tr>
<tr>
<td>Battaglia R.</td>
<td>486</td>
</tr>
<tr>
<td>Begus V.</td>
<td>330</td>
</tr>
<tr>
<td>Bellabdaoui A.</td>
<td>251/259</td>
</tr>
<tr>
<td>Beltram C.</td>
<td>333</td>
</tr>
<tr>
<td>Berthold M.R.</td>
<td>58</td>
</tr>
<tr>
<td>Biroukou A.</td>
<td>55</td>
</tr>
<tr>
<td>Blanzieri E.</td>
<td>55</td>
</tr>
<tr>
<td>Boussu F.</td>
<td>330</td>
</tr>
<tr>
<td>Brito A.C.</td>
<td>415</td>
</tr>
<tr>
<td>Brunetta L.</td>
<td>97</td>
</tr>
<tr>
<td>Bruniaux P.</td>
<td>365/370</td>
</tr>
<tr>
<td>Buffham B.A.</td>
<td>242</td>
</tr>
<tr>
<td>Caico R.</td>
<td>17</td>
</tr>
<tr>
<td>Carubelli A.</td>
<td>353</td>
</tr>
<tr>
<td>Carullo M.</td>
<td>74</td>
</tr>
<tr>
<td>Cascone A.</td>
<td>156</td>
</tr>
<tr>
<td>Castelfranchi C.</td>
<td>46</td>
</tr>
<tr>
<td>Catania V.</td>
<td>285</td>
</tr>
<tr>
<td>Ceanga E.</td>
<td>237</td>
</tr>
<tr>
<td>Cebron N.</td>
<td>58</td>
</tr>
<tr>
<td>Chalençon S.</td>
<td>359</td>
</tr>
<tr>
<td>Chatzinikolaou A.</td>
<td>229</td>
</tr>
<tr>
<td>Cichocka A.</td>
<td>365/370</td>
</tr>
<tr>
<td>Coen-Porisini A.</td>
<td>74</td>
</tr>
<tr>
<td>Coiro D.P.</td>
<td>115</td>
</tr>
<tr>
<td>Colloc J.</td>
<td>387/454</td>
</tr>
<tr>
<td>Comelli M.</td>
<td>449</td>
</tr>
<tr>
<td>Conley W.</td>
<td>163</td>
</tr>
<tr>
<td>Conti V.</td>
<td>41</td>
</tr>
<tr>
<td>Cossentino M.</td>
<td>17</td>
</tr>
<tr>
<td>Cugini U.</td>
<td>353</td>
</tr>
<tr>
<td>Cumming I.W.</td>
<td>242</td>
</tr>
<tr>
<td>de Brito E.R.</td>
<td>182</td>
</tr>
<tr>
<td>de Gijt M.</td>
<td>399</td>
</tr>
<tr>
<td>De Marco A.</td>
<td>115</td>
</tr>
<tr>
<td>de Salabert A.</td>
<td>493</td>
</tr>
<tr>
<td>Dhaene T.</td>
<td>297</td>
</tr>
<tr>
<td>Di Fatta G.</td>
<td>58</td>
</tr>
<tr>
<td>Di Nuovo A.G.</td>
<td>285</td>
</tr>
<tr>
<td>Dill F.</td>
<td>58</td>
</tr>
<tr>
<td>Doering S.</td>
<td>168</td>
</tr>
<tr>
<td>Dohmen L.</td>
<td>191</td>
</tr>
<tr>
<td>Durante T.</td>
<td>156</td>
</tr>
<tr>
<td>Edwards J.S.</td>
<td>479</td>
</tr>
<tr>
<td>Elia A.</td>
<td>277</td>
</tr>
<tr>
<td>Fabiano F.S.</td>
<td>290</td>
</tr>
<tr>
<td>Falcone R.</td>
<td>46</td>
</tr>
<tr>
<td>Féniès P.</td>
<td>449/459</td>
</tr>
<tr>
<td>Fortino G.</td>
<td>17</td>
</tr>
<tr>
<td>Fujarewicz K.</td>
<td>178</td>
</tr>
<tr>
<td>Gabriel T.R.</td>
<td>58</td>
</tr>
<tr>
<td>Gagli G.</td>
<td>86</td>
</tr>
<tr>
<td>Gallo I.</td>
<td>74</td>
</tr>
<tr>
<td>Galuszka A.</td>
<td>234</td>
</tr>
<tr>
<td>Garro A.</td>
<td>17</td>
</tr>
<tr>
<td>Genc A.</td>
<td>486</td>
</tr>
<tr>
<td>Gentile A.</td>
<td>86/290</td>
</tr>
<tr>
<td>Georg F.</td>
<td>58</td>
</tr>
<tr>
<td>Ghaeli M.</td>
<td>223</td>
</tr>
<tr>
<td>Ghorashi R.</td>
<td>508</td>
</tr>
<tr>
<td>Giorgini P.</td>
<td>49/55</td>
</tr>
<tr>
<td>Gómez-Sanz J.</td>
<td>9</td>
</tr>
<tr>
<td>Gorsich D.J.</td>
<td>103</td>
</tr>
<tr>
<td>Gorsi M.</td>
<td>431</td>
</tr>
<tr>
<td>Gouraud M.</td>
<td>449</td>
</tr>
<tr>
<td>Gries T.</td>
<td>340</td>
</tr>
<tr>
<td>Gunter D.</td>
<td>103</td>
</tr>
<tr>
<td>Haas R.</td>
<td>205</td>
</tr>
<tr>
<td>Habenicht I.</td>
<td>309</td>
</tr>
<tr>
<td>Hagestein J.</td>
<td>399</td>
</tr>
<tr>
<td>Hajiloo A.</td>
<td>504</td>
</tr>
</tbody>
</table>
### AUTHOR LISTING

<table>
<thead>
<tr>
<th>Author</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Happonen A</td>
<td>175</td>
</tr>
<tr>
<td>Hellgardt K</td>
<td>242</td>
</tr>
<tr>
<td>Hoffmann S</td>
<td>340</td>
</tr>
<tr>
<td>Hunt D</td>
<td>81</td>
</tr>
<tr>
<td>Huxford S</td>
<td>81</td>
</tr>
<tr>
<td>Ivanescu M</td>
<td>472</td>
</tr>
<tr>
<td>Janoušek V</td>
<td>69</td>
</tr>
<tr>
<td>Janssens G.K.</td>
<td>409</td>
</tr>
<tr>
<td>Joines J.A.</td>
<td>377</td>
</tr>
<tr>
<td>Kacur J</td>
<td>515</td>
</tr>
<tr>
<td>Karcher A</td>
<td>108</td>
</tr>
<tr>
<td>Kaveh B</td>
<td>508</td>
</tr>
<tr>
<td>Kay M.G.</td>
<td>377</td>
</tr>
<tr>
<td>Keshavarzi-Roonizi N.</td>
<td>508</td>
</tr>
<tr>
<td>King R.E.</td>
<td>377</td>
</tr>
<tr>
<td>Koncar V</td>
<td>365</td>
</tr>
<tr>
<td>Kostur K</td>
<td>515</td>
</tr>
<tr>
<td>Król likowski T</td>
<td>265</td>
</tr>
<tr>
<td>Krüger J</td>
<td>151</td>
</tr>
<tr>
<td>Kurec A</td>
<td>123</td>
</tr>
<tr>
<td>Kyosev Y.K.</td>
<td>340</td>
</tr>
<tr>
<td>La Spina E</td>
<td>486</td>
</tr>
<tr>
<td>Landaluze J</td>
<td>131</td>
</tr>
<tr>
<td>Lawrence P</td>
<td>300</td>
</tr>
<tr>
<td>Lee E</td>
<td>479</td>
</tr>
<tr>
<td>Lee P.L.</td>
<td>223</td>
</tr>
<tr>
<td>Lehmann T</td>
<td>108</td>
</tr>
<tr>
<td>Letherwood M.D.</td>
<td>103</td>
</tr>
<tr>
<td>Licht T</td>
<td>191</td>
</tr>
<tr>
<td>Lisounkin A</td>
<td>151</td>
</tr>
<tr>
<td>Lodewijks G (393/399/431)</td>
<td>437</td>
</tr>
<tr>
<td>Luczak H</td>
<td>191</td>
</tr>
<tr>
<td>Mabesa Jr. J.R.</td>
<td>123</td>
</tr>
<tr>
<td>Mana R.</td>
<td>353</td>
</tr>
<tr>
<td>Martínez F</td>
<td>131</td>
</tr>
<tr>
<td>Martínez-Esnola A.I.</td>
<td>131</td>
</tr>
<tr>
<td>Mascle C</td>
<td>210</td>
</tr>
<tr>
<td>Mason G</td>
<td>242</td>
</tr>
<tr>
<td>Mebrahtu H</td>
<td>217</td>
</tr>
<tr>
<td>Meinl T</td>
<td>58</td>
</tr>
<tr>
<td>Menna R</td>
<td>156</td>
</tr>
<tr>
<td>Meyer P</td>
<td>442</td>
</tr>
<tr>
<td>Milici G</td>
<td>41</td>
</tr>
<tr>
<td>Moerman S</td>
<td>431</td>
</tr>
<tr>
<td>Mönch L</td>
<td>314</td>
</tr>
<tr>
<td>Montagna S</td>
<td>25</td>
</tr>
<tr>
<td>Montés N</td>
<td>143</td>
</tr>
<tr>
<td>Moroni G</td>
<td>277</td>
</tr>
<tr>
<td>Nagy G</td>
<td>423</td>
</tr>
<tr>
<td>Nassehi V</td>
<td>508</td>
</tr>
<tr>
<td>Nicolás C.F.</td>
<td>131</td>
</tr>
<tr>
<td>Nicolosi F</td>
<td>115</td>
</tr>
<tr>
<td>Nikraz M</td>
<td>62</td>
</tr>
<tr>
<td>Nocent O</td>
<td>359</td>
</tr>
<tr>
<td>Nourrit J.-M.</td>
<td>359</td>
</tr>
<tr>
<td>Odelstad J</td>
<td>498</td>
</tr>
<tr>
<td>Ohl P</td>
<td>58</td>
</tr>
<tr>
<td>Oliva E</td>
<td>33</td>
</tr>
<tr>
<td>Omicini A</td>
<td>25/33</td>
</tr>
<tr>
<td>Ortega A</td>
<td>493</td>
</tr>
<tr>
<td>Ottjes J</td>
<td>399/431</td>
</tr>
<tr>
<td>Pacella M</td>
<td>277</td>
</tr>
<tr>
<td>Palesi M</td>
<td>285</td>
</tr>
<tr>
<td>Pana C</td>
<td>472</td>
</tr>
<tr>
<td>Pang Y</td>
<td>393</td>
</tr>
<tr>
<td>Patti D</td>
<td>285</td>
</tr>
<tr>
<td>Pavón J</td>
<td>9</td>
</tr>
<tr>
<td>Petitniot J.-L</td>
<td>330</td>
</tr>
<tr>
<td>Phillips D.G.</td>
<td>325</td>
</tr>
<tr>
<td>Pilato G</td>
<td>41/86</td>
</tr>
<tr>
<td>Piunti M</td>
<td>46</td>
</tr>
<tr>
<td>Polášek P</td>
<td>69</td>
</tr>
<tr>
<td>Porras J</td>
<td>175</td>
</tr>
<tr>
<td>Pujana-Arrese A</td>
<td>131</td>
</tr>
<tr>
<td>Ramaekers K</td>
<td>409</td>
</tr>
<tr>
<td>Reinbach I</td>
<td>340</td>
</tr>
<tr>
<td>Remion Y</td>
<td>359</td>
</tr>
<tr>
<td>Ricci A</td>
<td>25</td>
</tr>
<tr>
<td>Richardson D.J.</td>
<td>242</td>
</tr>
<tr>
<td>Rietveld L</td>
<td>139</td>
</tr>
<tr>
<td>Rizzi C</td>
<td>353</td>
</tr>
<tr>
<td>Robinson S</td>
<td>479</td>
</tr>
<tr>
<td>Rodier S</td>
<td>459</td>
</tr>
<tr>
<td>Roman S</td>
<td>217</td>
</tr>
<tr>
<td>Romanin-Jacur G</td>
<td>97</td>
</tr>
<tr>
<td>Rose O</td>
<td>319</td>
</tr>
<tr>
<td>Rovero G</td>
<td>333</td>
</tr>
<tr>
<td>Russell P.A.</td>
<td>242</td>
</tr>
<tr>
<td>Russo W</td>
<td>17</td>
</tr>
<tr>
<td>Sabov A</td>
<td>151</td>
</tr>
<tr>
<td>Author</td>
<td>Page</td>
</tr>
<tr>
<td>-----------------</td>
<td>------</td>
</tr>
<tr>
<td>Sansores C.</td>
<td>9</td>
</tr>
<tr>
<td>Santangelo A.</td>
<td>86</td>
</tr>
<tr>
<td>Santos F.C.G.</td>
<td>182</td>
</tr>
<tr>
<td>Sarfaraz A.R.</td>
<td>199</td>
</tr>
<tr>
<td>Sauer A.</td>
<td>168</td>
</tr>
<tr>
<td>Schlick C.</td>
<td>191</td>
</tr>
<tr>
<td>Schmidt L.</td>
<td>191</td>
</tr>
<tr>
<td>Schmied R.</td>
<td>108</td>
</tr>
<tr>
<td>Schreck G.</td>
<td>151</td>
</tr>
<tr>
<td>Schuh G.</td>
<td>168</td>
</tr>
<tr>
<td>Schulz J.</td>
<td>442</td>
</tr>
<tr>
<td>Sieb C.</td>
<td>58</td>
</tr>
<tr>
<td>Sinha K.</td>
<td>205</td>
</tr>
<tr>
<td>Sirbu G.</td>
<td>237</td>
</tr>
<tr>
<td>Skrzypczyk K.</td>
<td>234/467</td>
</tr>
<tr>
<td>Slaviček P.</td>
<td>69</td>
</tr>
<tr>
<td>Sorbello F.</td>
<td>41/290</td>
</tr>
<tr>
<td>Sorce S.</td>
<td>486</td>
</tr>
<tr>
<td>Stępień P.</td>
<td>265/269</td>
</tr>
<tr>
<td>Stoian E.</td>
<td>472</td>
</tr>
<tr>
<td>Stoian V.</td>
<td>472</td>
</tr>
<tr>
<td>Strange C.</td>
<td>217</td>
</tr>
<tr>
<td>Summons P.</td>
<td>387/454</td>
</tr>
<tr>
<td>Szatkiewicz T.</td>
<td>265</td>
</tr>
<tr>
<td>Termine F.</td>
<td>17</td>
</tr>
<tr>
<td>Thoney K.A.</td>
<td>377</td>
</tr>
<tr>
<td>Thron T.</td>
<td>423</td>
</tr>
<tr>
<td>Tornero J.</td>
<td>143</td>
</tr>
<tr>
<td>Tran C.D.</td>
<td>325</td>
</tr>
<tr>
<td>van der Heijden G.H.M.</td>
<td>325</td>
</tr>
<tr>
<td>Veeke H.P.M.</td>
<td>393/437</td>
</tr>
<tr>
<td>Viroli M.</td>
<td>33</td>
</tr>
<tr>
<td>Vitabile S.</td>
<td>41</td>
</tr>
<tr>
<td>Walker R.</td>
<td>217</td>
</tr>
<tr>
<td>Wassan N.</td>
<td>423</td>
</tr>
<tr>
<td>Weigl K.H.</td>
<td>5</td>
</tr>
<tr>
<td>Wiswedel B.</td>
<td>58</td>
</tr>
<tr>
<td>Worm I.</td>
<td>139</td>
</tr>
<tr>
<td>Zanzi A.</td>
<td>74</td>
</tr>
<tr>
<td>Zhou F.</td>
<td>345</td>
</tr>
<tr>
<td>Zimmermann J.</td>
<td>314</td>
</tr>
</tbody>
</table>