

# Mapping from Business Processes to Requirements Specification

**Svatopluk Štolfa, Ivo Vondrák**

Department of Computer Science, VŠB - Technical University of Ostrava,  
17.listopadu 15, Ostrava-Poruba, Czech Republic  
{svatopluk.stolfa, ivo.vondrak}@vsb.cz

**Abstract:** There are three basic reasons why you might need to model a business: to re-engineer a business, to improve a business process and to automate a business process. Nevertheless, another reason may be very useful for analyst of software systems and their customers – to understand and automatically generate functional requirements to the system. The present paper describes the process of redefinition of the business process captured by the activity diagram and its mapping to the requirements specification represented by the use case model.

## 1. Introduction

Business process is a set of one or more linked procedures or activities, which collectively realize a business objective or policy goal [12, 13]. Business process gives us an opportunity to manage project effectively by organization, simulation and realization of accurate planned processes. On the other hand, we can benefit from using business process for purposes of requirement specification. From this point of view, the main goal of the business process modeling is to provide common language for communities of software and business engineers. Unfortunately, this very important aspect of usage of business processes is not sufficiently covered by current technologies.

The goal of the requirements workflow is to describe what the system should do by specifying its functionality. Requirements modeling allows to the developer and the customer to agree with that description. For example, use case model examines the system functionality from the perspective of actors and use cases [2, 7, 14]. An actor is someone (user) or something (other system) that may interact with the system being developed. A use case is a pattern of behavior the system exhibits. Each use case is a sequence of related transactions performed by the actor and the system in a dialog. Use case model is described by UML use case diagrams. Use cases can be structured by “include”, “extends” and “generalization” relationships and actors by “generalization” relationship.

These are the basic definitions of business process and requirements specification represented by use case modeling mentioned in the title.

Some of software processes already refer to business modeling as a pre-requisition to the next steps of software development. Although, most of them do not clearly define the way how to use business processes, most of them can see the potential of business modeling for software development and want to be ready to introduce this process. Since the field of software engineering is changing so rapidly, inventors of these software processes must involve even unspecified solutions into their software processes to be able to compete in this field.

On the other hand, top-level managers/CEOs, CIOs, CTOs, planners, strategists and others are nowadays focused on business process management that should cover their needs of business. Business process management is an old discipline that allows you to model organizational structure, define the business processes and show the integration between them. Since today's business is closely tied to the application of appropriate software systems and its usage, business process management brings the business perspective and the IT infrastructure together. However, the concrete specification of usage or transformation of business process models into the software models is still missing.

In the following pages, it will be shown that business process may be also useful for mapping between business process models and use case models. The next sections briefly outline activity diagram as a semi-formal description of business process, introduces use case diagram and

the example. This will be followed by explanation of principles of mapping between business process and use case diagram. The final section will discuss the conclusions and future work that will be devoted to the formalization of this process.

## 2. Activity Diagram and Use Case Diagram Notation

In this section, activity and use case diagrams notations will be explained.

### 2.1 Activity Diagram

Business processes may be written down by various ways, such as business process modeling method or many others formal or semi-formal languages. For our purpose, the best way how to describe business process is UML activity diagram. States of business processes can be captured by activities and transitions between these states by transitions in activity diagram. Swim-lanes are used for organizing responsibility for activities provided by workers. Activity diagrams describe control flow given by inner mechanism of the execution.

### 2.2 Use Case Diagram

Basic description of the use case diagram has been mentioned above in the section introduction.

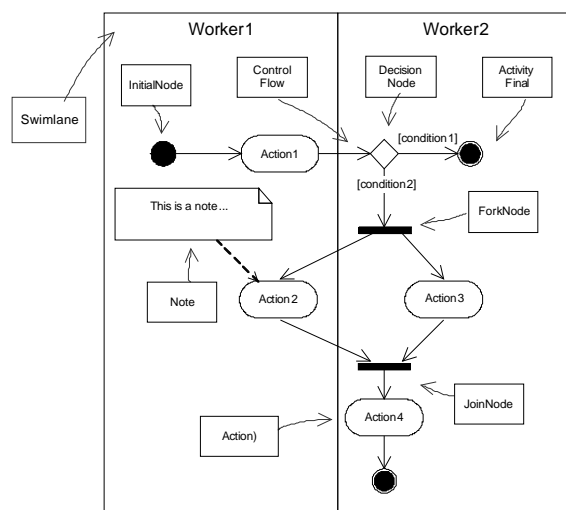


Fig. 1: Activity diagram notation

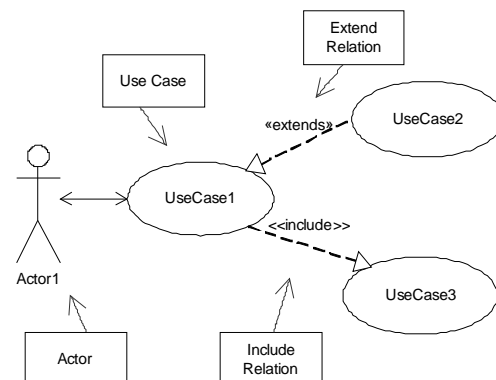


Fig. 2: Use case diagram notation

## 3. Business Process - Example

This section introduces the example that will be used for the explanation of the transformation process.

Figure 3 shows the two main processes in video lending library: “Lending” and “Returning”. First action is “customer enters”. If customer has something to return, he returns these items. Next symbol is “splitting transitions” and two control flows may be executed concurrently from this point. Left side is a part of lending process and right side is a part of returning process. Left part passes to “select”. If something was selected, customer must be found in records or must be registered. Then customer can borrow his items. Right side passes to “check items”, where damaged and late returned items are checked. If something was returned late or damaged, fine must be filled-in. Both parallel parts are joined. Next decision is concerning on payment. If customer has something to pay, action “payment” is invoked. Then customer leaves the lending office.

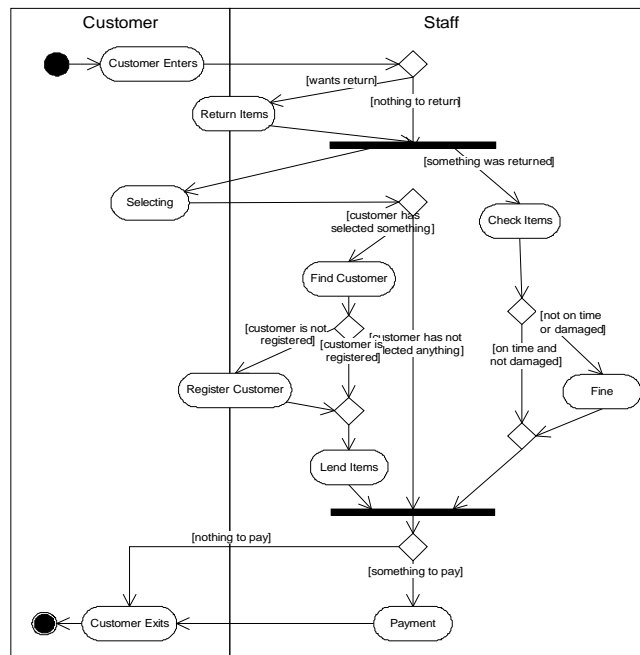


Fig. 3: Use Business process for video lending library

### 3. Mapping Activities to Use Cases

The transition from business process to use case diagram consist of several steps. At first, we should determine what actions will be provided by newly developed system and highlight them. Other actions remain human. Human actions are excluded and business process must be rewritten. In our example, actions Customer Enters, Return Items, Payment and Customer Exits will not be provided by the system.

Activities comprise both use cases and their components. It must be decided which of actions are the parts of use cases and which of them compose the use case as the whole. There are two possible ways how to compose use cases from actions and data flows. First off all, action may be transformed directly to use case. It is usually called one-to-one mapping. Another method used for use case creation is “mapping several actions to use cases”, where several actions are mapped to one general use case. That general use case will be the source use case for extend or include link to other use cases derived by “one-to-one mapping” or “mapping several actions to use cases” methods from subsets of original set of actions. Proper usage of these methods will be explained in the following sections.

### 4. Organize Use Cases

As mentioned above, UML use case diagram allows structuring use cases by relationships “extends”, “include” and “generalization”. Several basic mapping patterns have been developed to structure use cases according to the context of the original activities [9, 10]. Activities and selected parts of business processes are transformed to use cases and structured by mapping patterns.

Several main mapping patterns are used to organize use cases derived from activity diagram to use case diagram. Activity diagrams of these four patterns represent basic elements that may be composed together to create complex business process. It means that actions in patterns may be replaced by diagrams of other patterns. Let us to brief introduce Sequential Pattern, Optional Pattern as basic elements for activity diagram creation and backward recognition.

#### 4.1 Sequential Pattern

*Intent:* Organize use cases derived from sequential actions by “include” relationships.

Name of this pattern is derived from sequential execution of actions. Several actions are

mapped to one general use case using “mapping several actions to one use case” method. Sequential actions (or whole parts of other patterns that may replace these actions) are mapped to other use cases. Pattern uses “include” relationship to connect use cases, derived from sequential actions, to the main use case (see Fig. 4). The “include” relationship indicates that “UseCase1” uses (includes) “UseCase Action2” and “UseCase Action3”. Obviously, the number of sequential actions may be greater than two.

This pattern should be used for mapping of a set of sequential actions.

### 4.2 Optional Pattern

*Intent:* Organize use cases derived from optional actions by “extends” relationships.

This pattern shows how an optional action can be mapped and structured to use case diagram. The optional action can be enacted only if condition is true. Otherwise action is skipped (see Fig. 5). If a condition block and additional action are parts of one use case “UseCase1”, then “Action2” should be mapped to “UseCase Action2” which extends “UseCase1”. “Extends” relationship specifies that “UseCase Action2” may extend “UseCase1”.

This pattern should be used when optional action is mapped to use case and all around this action is a part of one use case.

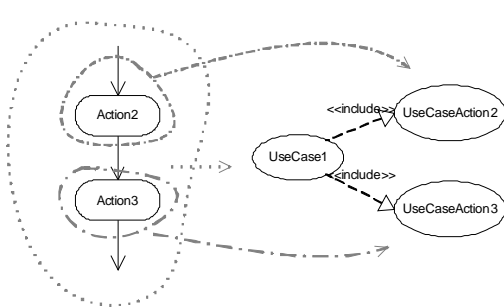


Fig. 4: Sequential pattern

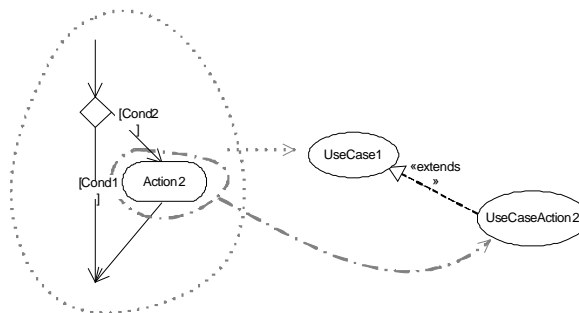


Fig. 5: Optional pattern

## 5. Process of Transformation

Input activity diagram for this process have to be composed using the basic elements defined by patterns or have to be redefined using this elements in allowed compositions described in the previous section. Only this well-formed activity diagram may be automatically transformed to use case diagram.

Let’s see the application of this mapping on our example. Final use case diagram was generated according to mapping patterns (see Fig. 6).

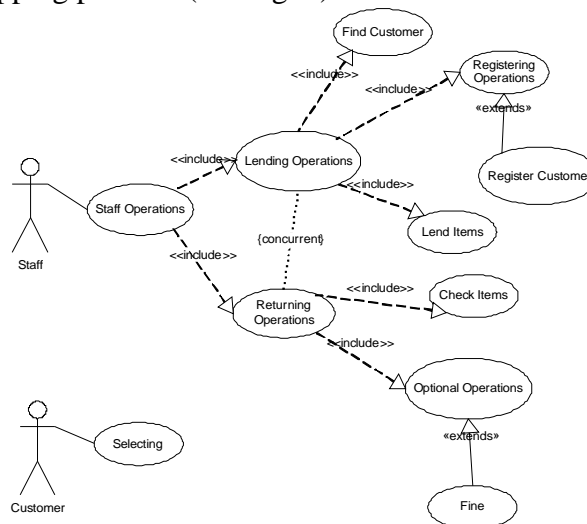


Fig. 6: Final use case diagram

## 6. Conclusions And Future Work

In this text, we have seen that business process modeling may be successfully used for definition of requirements specification (represented by the use case diagrams). I have briefly presented the transformation process on example. Even without precisely formal definition, this transformation process may be useful at least as the guide how to recognize useful parts of the business process management for the purposes of software requirements definition.

This article has only been able to touch on the most general features of the transformation process from business processes to use case diagrams. There are still unsolved problems such as transformation always depends on the right level of the business process abstraction that we have chosen. Another problem is, for example, if the business process is very bitty and all actions are mapped to use cases than it may cause into the granularity problem. A lot of use cases will be generated and even small software project may become obscure. Detailed behavior (e.g. what happen if login fails etc.) should be solved by use case alternative scenario paths, sequential diagrams may be generated etc. There are also pre-conditions such as business processes have to be prepared in well-formed form according to patterns or redefined to meet the pre-condition needs.

Clearly, further studies are needed to solve all possible problems, because business process management and requirement specification are very complex and this transformation process cannot solve all problems yet.

## 7. References

- [1] W.M.P. van der Aalst: Formalization and Verification of Event-driven Process Chains, Computing Science Reports 98/01, 1998
- [2] Booch, G., Rambough, J., Jacobson, I.: The Unified Modeling Language User Guide, Addison-Wesley, 1998
- [3] Champeaux, D., Lea, D., and Faure, P.: The process of object-oriented design, ACM SIGPLAN Notices (27:10), October 1992, pp. 45-62.
- [4] Chen, P.P. Ed.: Entity-Relationship Approach to Information Modeling and Analysis, Amsterdam, North-Holland, 1993
- [5] Fernandez, E. B.: Building systems using analysis patterns, Procs. 3rd Int. Soft. Architecture Workshop (ISAW3), Orlando, FL , November 1998 , 37-40
- [6] Fowler, M. Analysis Patterns: Reusable Object Models, Reading, MA: Addison-Wesley 1995
- [7] Schneider G., Winters J. P.: Applying Use Cases Second Edition - A Practical Guide, Addison-Wesley, 2001
- [8] Štolfa, S., Vondrák, I.: A Description of Business Process Modeling as a Tool for Definition of Requirements Specification. *In Proceedings of System Integration 2004*. Ed. Jan Pour, Prague:, 2004, p. 463-469, ISBN 80-254-0701-3
- [9] Štolfa, S., Vondrák, I.: An Explanation of Automatized Transformation Procedure from Business Processes to Use Case Diagrams. *In Proceedings of ISIM'04*. Ed. Miroslav Beneš, Rožnov pod Radhoš m:MARQ, 2004, 101-107, 80-85988-99-2
- [10]Štolfa, S., Vondrák, I.: Using Business Modeling Methods for Requirements Specification. *In Proceedings of SCI 2004*. Ed. Nagib Callaos, Vladimir Lefebvre, Elizabeth Hansen, Tilo Dickopp, Su Jaw-Sin, Orlando, FL:IIIS, 2004, vol. IV, P. 298-302, ISBN 980-6560-13-2
- [11]Štolfa, S., Vondrák, I.: Using the Business Process for Use Case Model Creation. *In ISIM '03*. Ed. M. Beneš, Ostrava, 2003, p. 129-137, ISBN 80-85988-84-4
- [12]Vondrák, I.: Methods for Business Process Modeling, Research Project, UNDP, 1997
- [13]Vondrák, I., Szturc, R., Kružel, M.: BPM – OO Metod for Business Process Modeling, *In ISM '99*, CSSS, Rožnov pod Radhoš m, ISBN 80-85988-31-3, pp. 155-163
- [14]<http://www.omg.org/technology/uml/index.htm>, *UML 2.0 Specification* (Final Adopted Specification), OMG, October 2004