Joseph Barjis (Ed.)

LNBIP 63

Enterprise and Organizational Modeling and Simulation

6th International Workshop, EOMAS 2010
held at CAiSE 2010, Hammamet, Tunisia, June 2010
Selected Papers

Springer

Lecture Notes in Business Information Processing 63

Joseph Barjis (Ed.)

Enterprise and Organizational Modeling and Simulation

6th International Workshop, EOMAS 2010
held at CAiSE 2010, Hammamet, Tunisia, June 7-8, 2010
Selected Papers

Springer

Volume Editor

Joseph Barjis
Delft University of Technology
Jaffalaan 5, 2628 BX Delft, The Netherlands
E-mail: j.barjis@tudelft.nl

Library of Congress Control Number: Applied for

ACM Computing Classification (1998): J.1, H.3.5, H.4.1, D.2

ISSN 1865-1348
ISBN-10 3-642-15722-X Springer Berlin Heidelberg New York
ISBN-13 978-3-642-15722-6 Springer Berlin Heidelberg New York

springer.com

Printed in Germany

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India
Printed on acid-free paper 06/3180 5 4 3 2 1 0

Preface

Enterprises of the 21st century are crucial components in delivering services to society and contributing to economic prosperity. Service is delivered when an enterprise is conducting its business within its business environment. With the growing complexity of modern business processes and continuously changing business environment, enterprise study (enterprise engineering) requires profound engineering approaches with properties such as ability for reengineering, scalability, adaptability, and reimplementation. Enterprises are purposefully designed and implemented systems to fulfill certain functions. As any system, enterprises are objects of continuous improvements, redesign and reimplementation. Usually, a redesigning activity is triggered by changes in the business environment, where the enterprise is functioning (delivering its service), or an internal need for efficiency. The departure point for any design or redesign activity pertinent to an enterprise is first to understand the enterprise business processes. Therefore, in the overall enterprise engineering activities, business process modeling plays a central role. However, an extended enterprise and organizational study involves both analysis and design activities, in which modeling and simulation play prominent roles. The growing role of modeling and simulation attracts serious attention of researchers in the context of enterprises. Modeling and simulation are the tools and methods that are effective, efficient, economic, and widely used in enterprise engineering, organizational study, and business process management. Complementary insights of modeling and simulation in enterprise engineering constitute a whole cycle of study of these complex sociotechnical system enterprises. In order to monitor and study the business processes and interaction of actors in a realistic and interactive environment, simulation has proven to be a powerful tool and method, especially if simulation is supported with rich animation and gaming elements. In order to explore these topics, address the underlying challenges, find and improve solutions, and demonstrate application of modeling and simulation in enterprise engineering, its organization and underlying business processes, peer-refereed papers were accepted for presentation at EOMAS 2010. A subset of these fully reviewed papers was selected for publication in this book in the LNBIP series published by Springer.

June 2010 Joseph Barjis

Organization

The EOMAS workshop is organized annually as an international forum for researchers and practitioners in the field of enterprise and organization modeling and simulation. Organization of this workshop and peer review of the contributions made to this workshop are accomplished by an international team of researchers in the field of enterprise modeling and simulation.

Workshop Organizers

Workshop Chair

Joseph Barjis	Delft University of Technology, The Netherlands

Program Co-chairs

Ghaith Rabadi	Old Dominion University, USA
Murali Mohan Narasipuram	City University of Hong Kong, China

Program Committee

Antonia Albani	Delft University of Technology, The Netherlands
Anteneh Ayanso	Brock University, Canada
Joseph Barjis	Delft University of Technology, The Netherlands
Ygal Bendavid	Polytechnic and Academia RFID, Canada
Peter Bollen	Maastricht University, The Netherlands
Mahmoud Boufaida	Mentouri University of Constantine, Algeria
Tatiana Bouzdine-Chameeva	BEM - Bordeaux Management School, France
Manuel I. Capel-Tuñón	University of Granada, Spain
Jan Dietz	Delft University of Technology, The Netherlands
Samuel Fosso Wamba	University of Wollongong, Australia
Jose Luis Garrido Bullejos	University of Granada, Spain
Ashish Gupta	Minnesota State University Moorhead, USA
Oleg Gusikhin	Ford Research and Advanced Engineering, USA
Johann Kinghorn	Stellenbosch University, South Africa
P. Radha Krishna	Infosys Technologies Ltd., India
Peggy Daniels Lee	Penn State Great Valley, USA
Oswaldo Lorenzo	Instituto de Empresa, Spain
Prabhat Mahanti	University of New Brunswick, Canada
Yuri Merkuryev	Riga Technical University, Latvia
Vojtech Merunka	Czech University of Life Sciences Prague, Czech Republic

Alta van der Merwe	University of South Africa, South Africa
Martin Molhanec	Czech Technical University in Prague, Czech Republic
Murali Mohan Narasipuram	City University of Hong Kong, China
Ghaith Rabadi	Old Dominion University, USA
Srini Ramaswamy	University of Arkansas at Little Rock, USA
Han Reichgelt	Southern Polytechnic State University, USA
Peter Rittgen	University College of Boras, Sweden
Mamadou Seck	Delft University of Technology, The Netherlands
Natalia Sidorova	Eindhoven University, The Netherlands
Michel Soares	Federal University of Uberlandia, Brazil
David Sundaram	The University of Auckland, New Zealand
Nick Szirbik	University of Groningen, The Netherlands
Yutaka Takahashi	Senshu University, Japan
Andreas Tolk	Old Dominion University, USA
José Tribolet	Technical University of Lisbon, Portugal
Alexander Verbraeck	Delft University of Technology, The Netherlands

Auxiliary Reviewers

Kawtar Benghazi Akhlaki	University of Granada, Spain
Michele Fumarola	Delft University of Technology, The Netherlands
Yilin Huang	Delft University of Technology, The Netherlands
Rick van Krevelen	Delft University of Technology, The Netherlands
Max Erik Rohde	The University of Auckland, New Zealand
Christian Stahl	Eindhoven University of Technology, The Netherlands

Sponsoring Institutions

- SIGMAS (**S**pecial **I**nterest **G**roup on **M**odeling And **S**imulation of the Association for Information Systems) - in collaboration

- SIGSIM (**S**pecial **I**nterest **G**roup on **S**imulation of the Association for Computing Machinery)

- CAiSE 2010 (International **C**onference on **A**dvanced **I**nformation **S**ystems Engineering)

- TU Delft (Delft University of Technology Department of Systems Engineering)

Table of Contents

Business Process Simulation Revisited

Wil M.P. van der Aalst

Department of Mathematics and Computer Science,
Eindhoven University of Technology,
P.O. Box 513, NL-5600 MB, The Netherlands
w.m.p.v.d.aalst@tue.nl

Abstract. Computer simulation attempts to "mimic" real-life or hypothetical behavior on a computer to see how processes or systems can be improved and to predict their performance under different circumstances. Simulation has been successfully applied in many disciplines and is considered to be a relevant and highly applicable tool in Business Process Management (BPM). Unfortunately, in reality the use of simulation is limited. Few organizations actively use simulation. Even organizations that purchase simulation software (stand-alone or embedded in some BPM suite), typically fail to use it continuously over an extended period. This keynote paper highlights some of the problems causing the limited adoption of simulation. For example, simulation models tend to oversimplify the modeling of people working part-time on a process. Also simulation studies typically focus on the steady-state behavior of business processes while managers are more interested in short-term results (a "fast forward button" into the future) for operational decision making. This paper will point out innovative simulation approaches leveraging on recent breakthroughs in process mining.

1 Limitations of Traditional Simulation Approaches

Simulation was one of the first applications of computers. The term "Monte Carlo simulation" was first coined in the Manhattan Project during World War II, because of the similarity of statistical simulation to games of chance played in the Monte Carlo Casino. This illustrates that that already in the 1940s people were using computers to simulate processes (in this case to investigate the effects of nuclear explosions). Later Monte Carlo methods were used in all kinds of other domains ranging from finance and telecommunications to games and workflow management. For example, note that the influential and well-known programming language Simula, developed in the 1960s, was designed for simulation. Simulation has become one of the standard analysis techniques used in the context of operation research and operations management. Simulation is particularly attractive since it is versatile, imposes few constraints, and produces results that are relatively easy to interpret. Analytical techniques have other advantages but typically impose additional constraints and are not as easy to use [9]. Therefore, it is no surprise that in the context of *Business Process Management* (BPM), simulation is one of the most established analysis techniques supported by a vast array of tools.

J. Barjis (Ed.): EOMAS 2010, LNBIP 63, pp. 1–14, 2010.

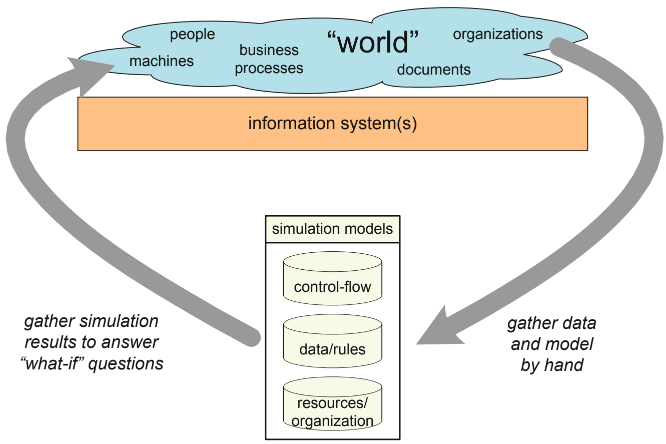


Fig. 1. Classical view on simulation: focus is on steady-state and model is made by hand

Figure 1 positions *business process simulation* in the context of a "world" supported by information systems. In the "world" consisting of people, organizations, products, processes, machines, etc. *information systems* play an increasingly dominant role. Moreover, there is continuous need for process improvements resulting in a better performance (e.g., better response times, less costs, higher service levels, etc.). Simulation can assist in this. Figure 1 shows the traditional use of simulation were data is gathered and used to parameterize hand-made models. These models are then used for simulation experiments answering "what-if" questions. For simulating business processes at least three *perspectives* need to be modeled: (a) *control-flow*, (b) *data/rules*, and (c) *resource/organization*. The control-flow perspective is concerned with the ordering of activities and uses design artifacts such as sequences, AND/XOR-splits/joins, loops, etc. [1]. The data/rules perspective models decisions made within the process and the role that data plays in these decisions. For simulation it is important not to model the data in too much detail and select the right abstraction level. The resource/organization perspective is concerned with the allocation of activities to resources, availability and speed of resources, and organizational boundaries [21]. In all of this *time* (e.g., the duration of an activity) and *probabilities* (e.g., the likelihood of following a particular path) play an important role. By answering "what-if" questions, managers and users get more insight into the effects of particular decisions.

Although many organizations have tried to use simulation to analyze their business processes at some stage, *few are using simulation in a structured and effective manner*. This may be caused by a lack of training and limitations of existing tools. However, as argued in this paper, there are also several additional and more fundamental problems. First of all, simulation models tend to *oversimplify* things. In particular the behavior of resources is often modeled in a rather naive manner. People do not work at constant speeds and need to distribute their attention over multiple processes. This can have dramatic effects on the performance of a process [2,15] and, therefore, such aspects should not be "abstracted away". Second, various *artifacts available are not used as input for simulation.* Modern organizations store events in logs and some may have accurate process models stored in their BPM/WFM systems. Also note that in may organizations, the state of the information system accurately reflects the state of the business processes supported by these systems because of the tight coupling between both. Today such information (i.e., event logs and status data) is rarely used for simulation or a lot of manual work is needed to feed this information into the model. Fortunately, *process mining* can assist in extracting such information and use this to realize performance improvements [4,7]. Third, the focus of simulation is mainly on "design" while managers would also like to use simulation for *"operational decision making"* (solving the concrete problem at hand rather than some abstract future problem). Fortunately, *short-term simulation* [16,20,24] can provide answers for questions related to "here and now". The key idea is to start all simulation runs from the current state and focus the analysis of the transient behavior. This way a "fast forward button" into the future is provided.

In the remainder, we elaborate on the above three problems and discuss some solution approaches grounded in process mining.

2 Oversimplified Simulation Models

Everything should be made as simple as possible,
but not one bit simpler.
Albert Einstein (1879-1955)

Simulation can be used to predict the performance under various circumstances, e.g., different business process re-engineering alternatives can be compared with the current situation. The value of such predictions stands or falls with the quality of the simulation model. Unfortunately, in many situations the quality of the simulation model leaves much to be desired. Basically, there are three problems: (a) the process is modeled incorrectly, (b) not enough data was collected to be able to parameterize the model, and (c) the language does not allow for the modeling of more subtle behaviors. The first two problems can be addressed by training people and a better validation of the model, e.g., by comparing the simulation results with real data. Here process mining can help as will be discussed in later sections. In this section, we focus on the last problem.

Probably the biggest problem of current business simulation approaches is that *human resources are modeled in a very naive manner*. As a result, it is not uncommon that the simulated model predicts flow times of minutes or hours while in reality flow times are weeks or even months. Therefore, we list some of the main problems encountered when modeling resources in current simulation tools. These problems stem from the fact that resources cannot be modeled adequately.

People are involved in multiple processes. In practice there are few people that only perform activities for a single process. Often people are involved in many different processes, e.g., a manager, doctor, or specialist may perform tasks in a wide range of processes. However, simulation often focuses on a single process. Suppose a manager is involved in 10 different processes and spends about 20 percent of his time on the process that we want to analyze. In most simulation tools it is impossible to model that a resource is only available 20 percent of the time. Hence, one needs to assume that the manager is there all the time and has a very low utilization. As a result the simulation results are too optimistic. In the more advanced simulation tools, one can indicate that resources are there at certain times in the week (e.g., only on Monday). This is also an incorrect abstraction as the manager distributes his work over the various processes based on priorities and workload. Suppose that there are 5 managers all working 20 percent of their time on the process of interest. One could think that these 5 managers could be replaced by a single manager (5*20%=1*100%). However, from a simulation point of view this is an incorrect abstraction. There may be times that all 5 managers are available and there may be times that none of them are available.

People do not work at a constant speed. Another problem is that people work at different speeds based on their workload, i.e., it is not just the distribution of attention over various processes, but also their absolute working speed that determines their capacity for a particular process. There are various studies that suggest a relation between workload and performance of people. A well-known example is the so-called Yerkes-Dodson law [23]. The Yerkes-Dodson law models the relationship between arousal and performance as a $\cap$-shaped curve. This implies that for a given individual and a given type of tasks, there exists an optimal arousal level. This is the level where the performance has its maximal value. Thus work pressure is productive, up to a certain point, beyond which performance collapses. Although this phenomenon can be easily observed in daily life, today's business process simulation tools do not support the modeling of workload dependent processing times.

People tend to work part-time and in batches. As indicated earlier, people may be involved in different processes. Moreover, they may work part-time (e.g., only in the morning). In addition to their limited availabilities, people have a tendency to work in batches (cf. Resource Pattern 38: Piled Execution [21]). In any operational process, the same task typically needs to be executed for many different cases (process instances). Often people prefer to let work-items related to the same task accumulate, and then process all of these in one batch. In most simulation tools a resource is either available or not, i.e., it is assumed that a resource is eagerly waiting for work and immediately reacts to any work-item that arrives. Clearly, this does not do justice to the way people work in reality. For example, consider how and when people reply to e-mails. Some people handle e-mails one-by-one when they arrive while others process their e-mail at fixed times in batch. Related is the fact that calendars and shifts are typically ignored in simulation tools. While holidays, lunch breaks, etc. can heavily impact the performance of a process, they are typically not incorporated in the simulation model.

Priorities are difficult to model. As indicated above, people are involved in multiple processes and even within a single process different activities and cases may compete for resources. One process may be more important than another and get priority. Another phenomenon is that in some processes cases that are delayed get priority while in other processes late cases are "sacrificed" to finish other cases in time. People need to continuously choose between work-items and set priorities. Although important, this is typically not captured by simulation models.

Process may change depending on context. Another problem is that most simulation tools assume a stable process and organization and that neither of them change over time. If the flow times become too long and work is accumulating, resources may decide to skip certain activities or additional resources may be mobilized. Depending on the context, processes may be configured differently and resources may be deployed differently. In [5] it is shown that such "second order dynamics" heavily influence performance.

The problems stem from oversimplified models. Note that although more than 40 resource patterns have been identified to describe the functionality of resource allocation mechanisms in the context of workflow management systems [21], few of these patterns are supported by today's business process simulation tools.

3 Learning from Event Logs

Learning is not compulsory ... neither is survival.
William Edwards Deming (1900-1993)

As discussed in the previous section, simulation models tend not to capture certain aspects or stick to an idealized variant of the real process. This can be partly addressed by better modeling techniques, e.g., additional parameters describing the resource characteristics. However, to adequately set these parameters and to make sure that processes are modeled accurately, we propose to also *exploit the information available in event logs.*

More and more information about (business) processes is recorded by information systems in the form of so-called "event logs" (e.g., transaction logs, audit trails, databases, message logs). As mentioned earlier, IT systems are becoming more and more intertwined with the processes they support, resulting in an "explosion" of available data that can be used for analysis purposes.

To illustrate the role that event logs can play, let us first explain Figure 2. We assume the existence of a collection of information systems that are supporting a "world" composed of business processes, people, organizations, etc. The *event data* extracted from such systems are the starting point for *process mining.* Note that Figure 2 distinguishes between *current data* and *historic data.* The former refers to events of cases (i.e., process instances) that are still actively worked on ("pre mortem"). The latter refers to events of completed cases, i.e., process

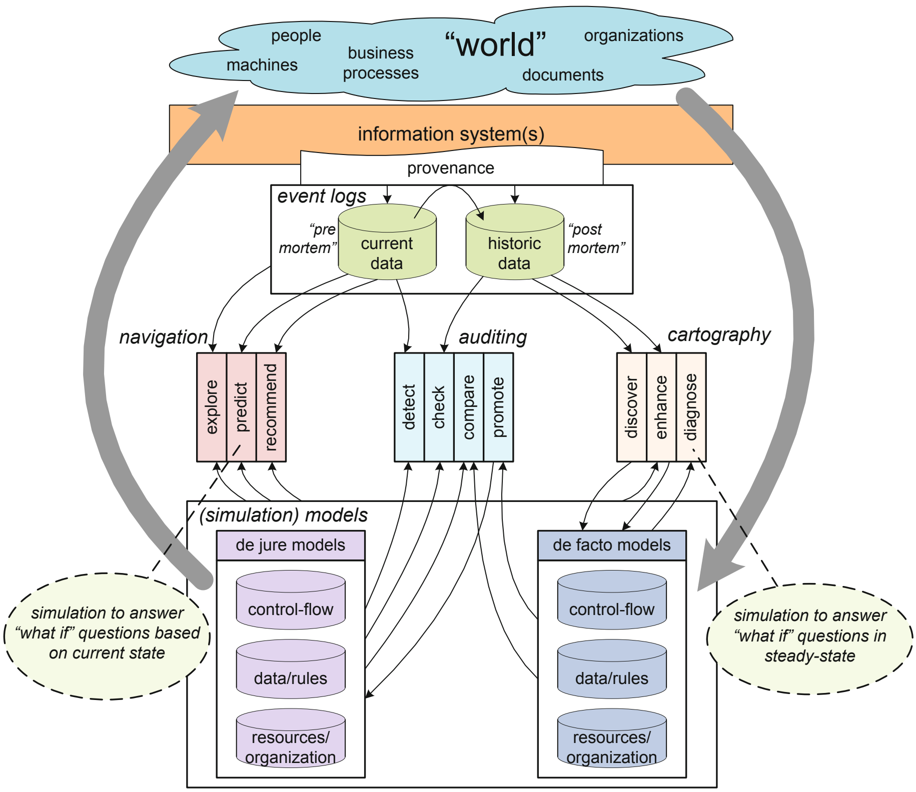


Fig. 2. Advanced business process simulation put into the context of process mining

instances that cannot be influenced anymore ("post mortem"). The historic data ("post mortem") can be any collection of events where each event refers to an instance (i.e., case), has a name (e.g., activity name), and has a timestamp. Note that some process mining techniques abstract from time. However, in the context of business process simulation these timestamps are of the utmost importance. The current data ("pre mortem") can be used to construct a well defined starting point for simulation. This is of particular importance for predictions in the near future.

The collection of event data is becoming more important. One the one hand, more and more event data are available. On the other hand, organizations depend on such data; not only for performance measurement, but also for auditing. We use the term *business process provenance* [10,11] to refer to the systematic collection of the information needed to reconstruct what has actually happened. The term signifies that for most organizations it is vital that "history cannot be rewritten or obscured". From an auditing point of view the systematic, reliable, and trustworthy recording of events is essential. Therefore, we propose to collect (whenever possible) provenance data outside of the operational information system(s) as shown in Figure 2. This means that events need to be collected and stored persistently. Note that semantics play an important role here, i.e., *events need to refer to a commonly agreed-upon ontology* [14].

The lower part of Figure 2 shows two types of models: *de jure models* are normative models that describe a desired or required way of working while *de facto models* aim to describe the actual reality with all of its intricacies (policy violations, inefficiencies, fraud, etc.). Both types of models may cover one or more perspectives and thus describe control-flow, time, data, organization, resource, and/or cost aspects. For process mining one can focus on a particular perspective. However, when the goal is to build simulation models all factors influencing performance need to be taken into account (e.g., when measuring utilization and response times, it is not possible to abstract from resources and focus on control-flow only). Simulation models can be based on a mixture of "de jure" and "de facto" information. The key idea of process mining is to not simply rely on de jure models that may have little to do with reality. Therefore, the goal is to shift more to "de facto models for simulation"; this will save time and increase quality.

In Figure 2 three main categories of activities have been identified: *cartography*, *auditing*, and *navigation*. The individual activities are briefly described below.

1. **Discover.** The discovery of good process models from events logs - comparable to geographic maps - remains challenging. Process discovery techniques can be used to discover process models (e.g., Petri nets) from event logs [4,7].
2. **Enhance.** Existing process models (either discovered or hand-made) need to be related to events logs such that these models can be enhanced by making them more faithful or by adding new perspectives based on event data. By combining historic data and pre-existing models, these models can be repaired (e.g., a path that is never taken is removed) or extended (e.g., adding time information extracted from logs).
3. **Diagnose.** Models (either de jure or de facto) need to be analyzed using existing model-based analysis techniques, e.g., process models can be checked for the absence of deadlocks or simulated to estimate cycle times. Probably the most widely used model-based analysis technique is simulation.
4. **Detect.** For on-line auditing, de jure models need to be compared with current data (events of running process instances) and deviations of such partial cases should to be detected at runtime. By replaying the observed events on a model, it is possible to do conformance checking while the process is unfolding.
5. **Check.** Similarly, historic "post mortem" data can be cross-checked with de jure models. For this conformance checking techniques are used that can pinpoint deviations and quantify the level of compliance [18].
6. **Compare.** De facto models can be compared with de jure models to see in what way reality deviates from what was planned or expected.
7. **Promote.** Based on an analysis of the differences between a de facto model and a de jure model, it is possible to promote parts of the de facto model to

a new de jure model. By promoting proven "best practises" to the de jure model, existing processes can be improved. For example, a simulation model may be improved and calibrated based on elements of a de facto model.

8. **Explore.** The combination of event data and models can be used to explore business processes. Here new forms of interactive process visualization can be used (visual analytics).
9. **Predict.** By combining information about running cases with models (discovered or hand-made), it is possible to make predictions about the future, e.g., the remaining flow time and the probability of success. Here simulation plays an important role. This will be elaborated in Section 4.
10. **Recommend.** The information used for predicting the future can also be used to recommend suitable actions (e.g. to minimize costs or time). The goal is to enable functionality similar to the guidance given by navigation systems like TomTom, but now in the context of BPM.

The first three activities are grouped under the term "cartography". Over time cartographers have improved their skills and techniques to create maps thereby addressing problems such as clearly representing desired traits, eliminating irrelevant details, reducing complexity, and improving understandability. Today, geographic maps are digital and of high quality. People can seamlessly zoom in and out using the interactive maps (cf. navigation systems like TomTom and services linked to Google Maps). Moreover, all kinds of information can be projected on these interactive maps (e.g., traffic jams, etc.). Process models can be seen as the "maps" describing the operational processes of organizations. Process mining techniques can be used to generate such maps. These maps can be simple and without executable semantics. However, as shown in [19] also simulation models can be discovered.

The next four activities are grouped under the term "auditing" as they compare normative/modeled behavior with real/recorded behavior. This does not involve simulation; however, these activities may help to increase the quality of discovered/hand-made simulation models.

The last three activities are grouped under the term "navigation". Navigation systems have proven to be quite useful for many drivers. People increasingly rely on the devices of TomTom, Garmin and other vendors and find it useful to *get directions* to go from A to B, know the *expected arrival time*, learn about *traffic jams* on the planned route, and be able to *view maps* that can be *customized* in various ways (zoom-in/zoom-out, show fuel stations, speed limits, etc.). However, when looking at business processes and their information systems, *such information is typically lacking.* Fortunately, a combination of process mining and simulation can help to provide navigation capabilities. The next section focuses on this.

4 Operational Support

If you don't know where you are going, any road will get you there.
Lewis Carroll (1832-1898)

Figure 2 illustrated that event logs can be used for all kinds of analysis, e.g., event logs can be used to discover and improve simulation models. In this section, we focus on *short-term simulation*, i.e., a detailed analysis of the near future based on the current state. Traditionally, business process simulation is mainly used for steady-state analysis and not for operational decision making. To explain the importance of short-term simulation, we first elaborate on the difference between *transient analysis* and *steady-state analysis*.

The key idea of simulation is to execute a model repeatedly. The reason for doing the experiments repeatedly, is to not come up with just a single value (e.g., “the average response time is 10.36 minutes”) but to provide confidence intervals (e.g., “the average response time is with 90 percent certainty between 10 and 11 minutes”). For transient analysis the focus is on the initial part of future behavior, i.e., starting from the initial state the “near future” is explored. For transient analysis the initial state is very important. If the simulation starts in a state with long queues of work, then in the near future flow times will be long and it may take some time to get rid of the backlog. For steady-state analysis the initial state is irrelevant. Typically, the simulation is started “empty” (i.e., without any cases in progress) and only when the system is filled with cases the measurements start.

Steady-state analysis is most relevant for answering strategic and tactical questions. Transient analysis is most relevant for operational questions. Lion’s share of contemporary simulation support aims at steady-state analysis and hence at strategic and tactical decision making. We advocate *more emphasis on simulation for operational decision making*. Therefore, we elaborate on short-term simulation and relate this to process mining and operational support.

Figure 3 shows the input used for operational support. *Historic data*, i.e., event logs, can be used to discover new models and to enhance existing models. This was already discussed in the previous section. The learned models can be combined with *current data* (i.e., states of cases and partial execution traces)

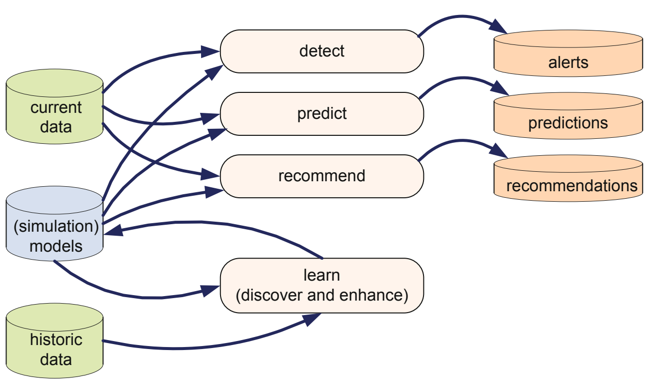


Fig. 3. Overview of operational support and the different types of data used

to *detect* deviations, *predict* performance, and to *recommend* decisions. Predictions may be based on regression models [12]. However, to predict more complex dynamic behavior, simulation can be used. In this paper, we distinguish between operational support at the *instance level* and at the *aggregate level*. The instance level focuses on a single case, e.g., a particular loan application that is being processed. It may be detected that the application is delayed and because of this an alert is generated. Moreover, for the partially executed loan application it may be predicted that the expected remaining processing time is two weeks and that therefore it is recommended to bypass an external credit check. Unlike recommendations and predictions at the instance level, operational support at the aggregate level is concerned with the whole process (or even a set of processes). Problems are now detected at the aggregate level ("response times are too long"). Moreover, predictions and recommendations are at the process level and do not refer to particular instances.

Table 1 provides examples of operational support questions. Both levels (instance level and aggregate level) are discussed in the remainder.

Operational support at the instance level. Figure 4 illustrates the three types of operational support. Starting point is some model and a partial trace. Note that the model is typically learned using classical process mining techniques. The partial trace refers to a case that is running. The left-hand side of Figure 4 shows a partial trace $\langle A, B \rangle$. Although Figure 4 does not show timestamps, resources, data, etc., these may be relevant for operational support.

For the case shown in Figure 4, we know that A and B occurred, but we do not know its future. Suppose now that the partial trace $\langle A, B \rangle$ is not possible according to the model. In this case, the operational support system should generate an alert. Another possibility would be that B took place three weeks after A while this should happen within one week. In such a case another notification could be sent to the responsible case manager. Such scenarios correspond to the *check* activity mentioned before. Figure 4 also illustrates the goal of *predictions*. Given the current state of a case, the model is used to make some kind of prediction [3,6]. For example, given the $\langle A, B \rangle$ trace it could be predicted that the remaining processing time is ten days. This prediction would be based on historic information both in the partial trace and in the event log used to learn the model. For the actual prediction a simple regression model can be used. However, for more complex scenarios, *short-term simulation* is a more likely option. Predictions are not restricted to time, but can also refer to costs, probability of a particular outcome, resource availability, etc. Closely related to predictions are *recommendations* [3,22]. The main difference is that recommendations suggest the next action based on possible continuations of the case. Based on the model, one can try all possible actions and see which one would lead to the best (predicted) performance. Note that recommendations are not only used for determining the next task, but also for allocating resources to work-items or for timing a particular action.

Table 1. Examples of various types of operational support at the instance level and the aggregate level

type of operational support	*instance level*	*aggregate level*
detect	Partially executed cases are monitored. As soon as a deviation occurs (e.g., a task is skipped or too late) an alert is given.	Processes are monitored as a whole and as soon as a deviation occurs (e.g., the average response times are too high or too many cases are in the pipeline) an alert is given.
predict	Predictions are made for specific cases, e.g., after each step the expected remaining processing time of the case is given. Predictions may also refer to costs and quality, e.g., the likelihood of succes for a particular instance. Short-term simulation can be used to generate such instance-level predictions.	Predictions are made for one process or a collection of processes. For example, it is predicted what the average flow time will be in the next two weeks. Predictions at the aggregate level may also refer to utilization ("How busy will people be next week?"), costs ("Will we reach the break-even point in this quarter?"), service levels, etc.
recommend	Predictions at the instance level can be turned into recommendations by exploring the effect of various decisions. For example, different routing choices can be simulated to predict the effect of such choices. Similarly, the effect of various allocation choices can be compared using simulation.	Predictions at the aggregate level can be used to generate recommendations. The effect of each decision can be analyzed using short-term simulation. For example, it may be recommended to temporarily hire two additional workers to avoid excessive waiting times.

Operational support at the aggregate level. In Figure 4 analysis is done at the instance level. However, many operational decisions transcend the level of an individual case. Decisions like temporarily adding two workers or stimulate overwork are made at the level of one or more processes rather than a single case. Short-term simulation is particularly useful for predictions at the aggregate level. Here, simple regression models are unable to capture queueing effects, dependencies, and typical work patterns.

Short-term simulation starts from the current state [16,20,24]. When a process-aware information system is present, it is relatively easy to extract the current state from the system and to upload this into the simulation model. By modifying the simulation model, various "what-if" scenarios can be investigated. For example, one can add or remove resources, skip activities, etc. and see what the effect is. Because the simulation experiments for these scenarios start from the

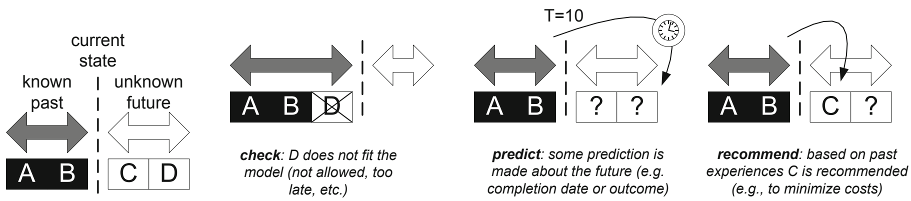


Fig. 4. Operational support at the instance level [3]

current state of the actual system, they provide a kind of "fast-forward button" showing what will happen in the near future, to support operational decision making. For instance, based on the predicted system behavior, a manager may decide to hire more personnel or stop accepting new cases.

5 Conclusion and Further Reading

The goal of this keynote paper is to provide a *critical analysis of the mainstream simulation approaches* for process management. On the one hand, the paper is based on practical experiences in numerous simulation projects (cf. [17] for examples). These experiences showed amongst others that it is almost impossible to adequately model resources in contemporary simulation tools. On the other hand, various process mining projects showed that reality rarely matches the expectations of the modeler. Models tend to describe idealized/unrealistic views on the business processes at hand. These practical experiences with simulation and process mining resulted in a better understanding of the pitfalls of traditional business process analysis. Some of the lessons learned have been reported. Moreover, as shown, business process simulation can benefit from *recent breakthroughs in process mining.*

Several of the ideas presented in this paper have been realized in the context of *ProM* (`www.processmining.org`, [8]) and *YAWL* (`www.yawlfoundation.org`, [13]). To conclude this paper, we provide pointers to papers detailing these results.

In [3] a concrete approach to operational support is given. This has been implemented in ProM and time-based predictions and recommendations are given by learning a transition system annotated with time information [6]. The focus in [3] is restricted to individual cases and temporal aspects.

In [15] it is shown how event logs can be used to learn about the behavior of people. For example, through process mining one can find empirical evidence for the Yerkes-Dodson law [23] and parameterize the corresponding simulation models.

ProM provides comprehensive support for the automated discovery of simulation models based on event logs. In [19] it is shown how different perspectives can be discovered and merged into one overall simulation model.

While the focus in [19] is on simulation models for steady-state analysis, the focus of [20] is on short-term simulation, i.e., transient analysis. This is achieved by an integration of ProM and YAWL. The workflow model, event log, and current state information provided by the workflow system YAWL are used by ProM to generate simulation models. These models are simulated using CPN Tools. Key element is that the simulation model is called continuously while using the latest state information. This way a "fast-forward button" is added to YAWL that allows users and manager explore the near future.

One of the key problems when using business process simulation is the fact that it is unrealistic to assume that people are continuous available. Availability and work-speed are fluid. As shown in [2], it is important to capture and parameterize this "fluidity" as it has a dramatic effect on flow times, etc.

The papers mentioned above present innovations in business process simulation. Although quite some work has been done in the context of ProM and YAWL, it remains crucial to further improve techniques and tools to better capture faithful simulation models. Hopefully this will stimulate more organizations to reap the benefits of business process simulation.

Acknowledgments. The author would like to thank all the people that contributed to the development of ProM and YAWL. This paper refers to simulation techniques developed together with Joyce Nakatumba, Anne Rozinat, Moe Wynn, Ronny Mans, Minseok Song, and several others.

References

1. van der Aalst, W.M.P., ter Hofstede, A.H.M., Kiepuszewski, B., Barros, A.P.: Workflow Patterns. Distributed and Parallel Databases 14(1), 5–51 (2003)
2. van der Aalst, W.M.P., Nakatumba, J., Rozinat, A., Russell, N.: Business Process Simulation: How to get it right? In: vom Brocke, J., Rosemann, M. (eds.) Handbook on Business Process Management, International Handbooks on Information Systems, pp. 317–342. Springer, Berlin (2010)
3. van der Aalst, W.M.P., Pesic, M., Song, M.: Beyond Process Mining: From the Past to Present and Future. In: Pernici, B. (ed.) CAiSE 2010. LNCS, vol. 6051, pp. 38–52. Springer, Heidelberg (2010)
4. van der Aalst, W.M.P., Reijers, H.A., Weijters, A.J.M.M., van Dongen, B.F., de Medeiros, A.A.K., Song, M., Verbeek, H.M.W.: Business Process Mining: An Industrial Application. Information Systems 32(5), 713–732 (2007)
5. van der Aalst, W.M.P., Rosemann, M., Dumas, M.: Deadline-based Escalation in Process-Aware Information Systems. Decision Support Systems 43(2), 492–511 (2007)
6. van der Aalst, W.M.P., Schonenberg, M.H., Song, M.: Time Prediction Based on Process Mining. BPM Center Report BPM-09-04, BPMcenter.org (2009)
7. van der Aalst, W.M.P., Weijters, A.J.M.M., Maruster, L.: Workflow Mining: Discovering Process Models from Event Logs. IEEE Transactions on Knowledge and Data Engineering 16(9), 1128–1142 (2004)
8. van der Aalst, W.M.P., van Dongen, B., Günther, C.W., Rozinat, A., Verbeek, E., Weijters, T.: ProM: The Process Mining Toolkit. In: de Medeiros, A.K.A., Weber, B. (eds.) Business Process Management Demonstration Track (BPMDemos 2009). CEUR Workshop Proceedings, vol. 489, pp. 1–4. CEUR-WS.org (2009)

9. Buzacott, J.A.: Commonalities in Reengineerd Business Processes: Models and Issues. Management Science 42(5), 768–782 (1996)
10. Curbera, F., Doganata, Y., Martens, A., Mukhi, N., Slominski, A.: Business Provenance: A Technology to Increase Traceability of End-to-End Operations. In: Meersman, R., Tari, Z. (eds.) OTM 2008, Part I. LNCS, vol. 5331, pp. 100–119. Springer, Heidelberg (2008)
11. Davidson, S., Cohen-Boulakia, S., Eyal, A., Ludaescher, B., McPhillips, T., Bowers, S., Anand, M., Freire, J.: Provenance in Scientific Workflow Systems. Data Engineering Bulletin 30(4), 44–50 (2007)
12. van Dongen, B.F., Crooy, R.A., van der Aalst, W.M.P.: Cycle Time Prediction: When Will This Case Finally Be Finished? In: Meersman, R., Tari, Z. (eds.) OTM 2008, Part I. LNCS, vol. 5331, pp. 319–336. Springer, Heidelberg (2008)
13. ter Hofstede, A.H.M., van der Aalst, W.M.P., Adams, M., Russell, N.: Modern Business Process Automation: YAWL and its Support Environment. Springer, Berlin (2010)
14. de Medeiros, A.A.K., van der Aalst, W.M.P.: Process Mining towards Semantics. In: Dillon, T.S., Chang, E., Meersman, R., Sycara, K. (eds.) Advances in Web Semantics I. LNCS, vol. 4891, pp. 35–80. Springer, Heidelberg (2008)
15. Nakatumba, J., van der Aalst, W.M.P.: Analyzing Resource Behavior Using Process Mining. In: Rinderle-Ma, S., Sadiq, S., Leymann, F. (eds.) BPM 2009. LNBIP, vol. 43, pp. 69–80. Springer, Berlin (2010)
16. Reijers, H.A., van der Aalst, W.M.P.: Short-Term Simulation: Bridging the Gap between Operational Control and Strategic Decision Making. In: Hamza, M.H. (ed.) Proceedings of the IASTED International Conference on Modelling and Simulation, pp. 417–421. IASTED/Acta Press, Anaheim, USA (1999)
17. Reijers, H.A., van der Aalst, W.M.P.: The Effectiveness of Workflow Management Systems: Predictions and Lessons Learned. International Journal of Information Management 25(5), 458–472 (2005)
18. Rozinat, A., van der Aalst, W.M.P.: Conformance Checking of Processes Based on Monitoring Real Behavior. Information Systems 33(1), 64–95 (2008)
19. Rozinat, A., Mans, R.S., Song, M., van der Aalst, W.M.P.: Discovering Simulation Models. Information Systems 34(3), 305–327 (2009)
20. Rozinat, A., Wynn, M., van der Aalst, W.M.P., ter Hofstede, A.H.M., Fidge, C.: Workflow Simulation for Operational Decision Support. Data and Knowledge Engineering 68(9), 834–850 (2009)
21. Russell, N., van der Aalst, W.M.P., ter Hofstede, A.H.M., Edmond, D.: Workflow Resource Patterns: Identification, Representation and Tool Support. In: Pastor, Ó., Falcão e Cunha, J. (eds.) CAiSE 2005. LNCS, vol. 3520, pp. 216–232. Springer, Heidelberg (2005)
22. Schonenberg, H., Weber, B., van Dongen, B.F., van der Aalst, W.M.P.: Supporting Flexible Processes Through Recommendations Based on History. In: Dumas, M., Reichert, M., Shan, M.-C. (eds.) BPM 2008. LNCS, vol. 5240, pp. 51–66. Springer, Heidelberg (2008)
23. Wickens, C.D.: Engineering Psychology and Human Performance. Harper, New York (1992)
24. Wynn, M., Rozinat, A., van der Aalst, W.M.P., ter Hofstede, A.H.M., Fidge, C.: Chapter 17: Process Mining and Simulation. In: Modern Business Process Automation: YAWL and its Support Environment, pp. 437–457. Springer, Berlin (2010)

The Relevance of Modeling and Simulation in Enterprise and Organizational Study

Joseph Barjis and Alexander Verbraeck

Jaffalaan 5, 2628 BX Delft
The Netherlands
{J.Barjis,A.Verbraeck}@TUDelft.NL

Abstract. Research on socio-technical systems, to which an enterprise, its organization, business processes, and supporting ICT belong, has been witnessing a resurging interest. Many research initiatives have been launched for the development of concepts, methods, and tools for the analysis and design of the enterprise structure, function, and processes, and for identification of actor roles and responsibilities in a consistent manner. One of the main drivers pushing research into this direction is the changing environment in which enterprises are functioning. In view of these trends, adoption of modeling and simulation, as two complementary tools for design, redesign, and improvement of enterprises, is becoming a standard practice. Especially in the face of ever evolving and changing business environment. In this article, we explain the relationship between enterprise, organization, and business processes on the one hand, and the relevance of modeling and simulation as a method in enterprise and organizational study.

Keywords: enterprise modeling, enterprise simulation, organizational modeling, organizational simulation, business process modeling, business process simulation, socio-technical systems.

1 Introduction

In the complex web of interrelated business processes, rules and procedures, and information and communication technologies (ICT), enterprises can no longer be regarded in an isolated manner as they are highly connected through networks, making the relationships increasingly complex and dynamic. Enterprises, their organization, business processes, and supporting ICT must be understood as socio-technical systems that consist of people (human actors), technical subsystems and their complicated relationships. In designing, redesigning, and improving such systems, modeling and simulation methods are not only relevant, but essential.

Modeling, especially during the analysis and design phase, a plays crucial role in any system development activity as it represents a design artifact in a more visualized manner such as intuitive diagrams (Shannon, 1975). The imperative role of modeling in enterprise study is in creating shared understanding and communicating design ideas and concepts among the stakeholders (analysts, users, decision makers). In this regard, simulation deals with comparison of different scenarios and possible design

J. Barjis (Ed.): EOMAS 2010, LNBIP 63, pp. 15–26, 2010.

ideas to investigate the solution space and capture the dynamic system behavior over time (Zeigler et al., 2000).

When viewed in a more extended way, important aspects for studying enterprise modeling and simulation are business processes, information systems, organizational change, IT impact, business-IT alignment, business strategy, etc. (Lorenzo & Diaz, 2005). Modern enterprises are supported and enabled by complex information systems or information and communication technologies in general. In turn, information systems also do not operate in isolation; they are designed, developed, and deployed in specific organizational contexts (settings). They are designed for certain objectives and tied to particular organizational processes (situations), e.g., order processing, product development, and process management.

An organization, as defined in the literature (Mintzberg, 1981; Scott, 2002; Dietz, 2008), is an arrangement of human actors purposefully organized to carry out a certain mission. Hence, an organization is a social system with human actors as its elements. Through the purposeful task execution, actions and interaction of these actors, business processes evolve. In modern enterprises, these interactions and processes are supported, linked and enacted via complex information systems such as enterprise information systems, e.g., enterprise resource planning systems, human resource information systems, or accounting information systems. For the design and redesign of these complex processes and systems, modeling and simulation are becoming increasingly popular (Harrison, 2002; Lorenzo & Diaz, 2005; Seila, 2005), but in practice they are far from becoming a standard tool.

While simulation has evolved into a mature discipline in other fields of engineering, such as manufacturing (Law & McComas, 1999; Miller & Pegden, 2000), military (Smith, 1998), transport (Brunner et al., 1998), etc., enterprise engineers and business process analysts have witnessed a much lower level of application of modeling and simulation (Melão & Pidd, 2003). This rather limited popularity of simulation can be explained by a number of challenges that can be classified into several dimensions. The categorization that follows has been compiled from a wide variety of sources (Giaglis et al., 1999; Carley, 2002; Lorenzo & Diaz, 2005). Often the same challenges are discussed in more than one source, therefore these challenges have been categorized using more general dimensions. These dimensions not only represent challenges currently hindering a more extensive use of enterprise modeling and simulation, but at the same time highlight motivations and potential agenda for research and development of new methods, tools, and approaches.

Here is a rather brief discussion of each dimension that will be elaborated later in the paper.

Conceptual Dimension: Today, there are marvelous simulation tools available in the market. No doubt that each has tremendous capability. The challenge of simulation is not in the tools but in how to carry out the simulation study in the right way (Balci, 1990, Carson, 2005). Only a conceptually well-designed model will result in a simulation study that yields success (Robinson, 2008). Despite its importance for simulation success, conceptual modeling is poorly studied and understood (Brooks, 2006). Furthermore, there are certain quality aspects that conceptual modeling should adhere to as well, for example pragmatic quality in the form of supporting automatic analysis and simulation (Barjis, 2008). However, for designing a rigorous conceptual model, one needs a thorough understanding of the concepts comprising enterprise and organizational study. These concepts are based on the understanding of enterprises as a complex socio-technical phenomenon.

Complexity Dimension: Enterprise business processes are scattered, interrelated, and complex. The enterprise activities are organized around business processes (process-centric design) as opposed to function. For example, the ordering process, shipping process, handling process, and billing process are closely interrelated processes, while at the same time each of these processes is owned by a separate business unit (purchasing department, delivery department, billing department). In many modern enterprises, processes such as billing and shipping may be partially or completely outsourced (Qureshi et al. 2007), which further adds to the complexity of business processes, their relationships, and interactions.

In the future, these processes will further grow in complexity and agility, which offers a promising role for modeling and simulation in their design and redesign. For example, as quoted in (van der Aalst, 2007), the popular enterprise resource planning system SAP consists of 604 event-driven process chains that models the underlying business processes supported by the R/3 system. One can imagine the number of models and sub-models a complex system like SAP might involve.

Social Dimension: Enterprise processes are socially dominated as these processes extensively involve human interactions. Thus, communication and interaction between the actors are an important aspect in the modeling and simulation of these processes. Only integrated modeling and simulation that includes both interactions (communication and coordination among actors) and actions (activities and processes) can realistically facilitate analysis, design and redesign of enterprise business processes.

These dimensions, mentioned rather for illustration purpose, are encompassing the main challenges in enterprise modeling and simulation. Developing a more complete taxonomy of these challenges is a subject of ongoing research, which is beyond the scope of this paper.

2 Conceptual Dimension: Enterprise, Organization, Business Process

For a better understanding of the interrelationship between an enterprise, its organization and business processes, we start by defining what we mean by an "enterprise" and "organization". From the myriad of definitions, we explain the notions of an enterprise and organization using the enterprise ontology theory. According to the enterprise ontology theory of Dietz (2008), the collective services that an enterprise provides to its environment are called the *business* of the enterprise. This definition represents the functional perspective of an enterprise (black box approach). The collective activities of an enterprise in which these services are delivered, along with the persons that carry out these activities, are called the *organization* of an enterprise. This definition represents the constructional perspective of an enterprise (white box approach). That is, an organization is a social arrangement of roles and responsibilities of persons and rules and norms by which the actions of these persons are governed to achieve certain goals. Thus, an organization is an artifact, formed and aligned according to the business goals of its enterprise.

A *business process* is a set of activities within an enterprise with a structure describing their logical order and dependence whose objective is to produce a desired result (Aguilar-Saven, 2004). Each service in an enterprise is delivered through one or more business processes, and each process is comprised of a series of activities. Collectively, business processes and their interrelationships, that enable to deliver service to customers or produce goods, constitute the enterprise *business system*.

As these notions imply, an *enterprise* has an organization (organizational processes) and a business system (business processes), both created for certain purposes. An enterprise provides services or delivers goods to its environment, thus, fulfilling its societal function. In this era of digital economy, the function of an enterprise and its organizational processes (interaction, coordination) are supported by complex information and communication technologies enabling delivery of the services in an effective and efficient way.

All this interrelationship of business processes, organizational processes, and supporting ICT establishes that a modern enterprise is a complex socio-technical system, whose analysis and design requires adequate and profound methods and tools stemming from rigorous theoretical foundations.

3 Complexity Dimension: Interwoven Business Processes and Workflows

There are many factors attributing to the complexity of the business system of modern enterprises. Firstly, enterprises are more and more moving towards process-centric design, where resources are aligned along the process flow. This leads to the second factor: since enterprises are becoming process centric, inter-departmental and inter-organizational workflows are becoming more complex and interwoven. Thirdly, many non-essential processes (services) are outsourced. For example, while order processing and packaging processes take place within the enterprise, shipping and delivery processes are carried by third party logistics; while patients are treated in the hospital, their laboratory samples might be sent to a third party for analysis. In cross-national business processes, additional polices and rules apply to the execution of different business processes. Increased fragmentation due to, market liberalization, and service-oriented architecture creates an extremely complex workflow within enterprises and between enterprises. Our focus of complexity is on business processes, but the underlying information technology, supporting modern enterprises, is also becoming more complex and distributed.

4 Social Dimension: Interaction

An enterprise is first of all a social system, where persons interact in an organized manner to fulfil the enterprise mission. As Cho et al. (1998) suggest, business process modeling should focus on interacting behaviour among people. This suggestion brings the social dimension of an enterprise to the forefront and makes it the focal point for enterprise modeling in terms of capturing human interactions. In accord with this position, Brandt et al. (1999) suggest an even more pragmatic approach where they state that an advanced simulation strategy should combine Human-In-The-Loop-Simulation and, for example, Petri Net Simulation to capture human involvement as well as activities. This suggests that a successful enterprise and organizational methodology for modeling and simulation should be capable of developing models based on different paradigms such as the Discrete Event Paradigm, the System Dynamics Paradigm, and the Agent-Based Paradigm.

5 Methods and Techniques of Study

What kind of methods and techniques do the analysts use in enterprise and organizational study? To answer this question, one should know the perspective and purpose of the study first. The perspective we take is of Systems Engineering (system analysis and design) and the purpose is to understand the enterprise's complex socio-technical setting, to analyze the current way of working, to visualize and quantitatively predict the effects of proposed or planned changes, and to compare various design options. In addition, the method choice depends on what aspect of the object of the study is the focal point, e.g., performance, engineering, reengineering, evaluation, etc. In any case, assuming a specific perspective and purpose narrows the set of choices to certain methods and techniques, as illustrated in the following list.

Prototyping method: In cases where rapid development of a system is required and some requirements of the system are defined, prototyping of the envisioned system might be a suitable starting point (Arnowitz et al., 2007). Furthermore, this method is popular when prototyping is the only way to communicate ideas, study initial behavior of the system, and complete the requirements. Most importantly, this method is used when a system can be developed on a smaller scale. However, a prototype not always allows full experimentation with all parameters of the system. Consequently, it is difficult to get a full picture of the system's dynamic behavior. Prototyping can be expensive and time consuming, especially when there are many design options.

Analytical method: If the study involves a performance study or capacity analysis where the input data are certain, analyst can develop mathematical models of the system (Aris, 1994; Gershenfeld, 1998). In this case, an abstraction of the system is designed and the model is studied by calculating its output parameters for different input data. This method is suitable when the input pattern (e.g., arrival of patients for service, placing of orders into the system, submission of insurance claims) has a clear distribution function. However, in many situations, input patterns do not adhere to a well-known distribution function and assumptions may lead to wrong output calculations (Hancock & Walter, 1979).

Modeling method: In the current practice of enterprise and organizational study, a very popular approach is modeling, especially modeling based on techniques allowing to draw static pictures using diagrams and then study the diagrams (Larkin et al., 1987; Koehler et al., 2008) such as IDEF (Integration DEFinition) (Mayer et al., 1992; IDEF, 2008), UML (Unified Modeling Language) (Jacobson et al., 1998; Booch et al., 1999; Torchiano & Bruno, 2003), EPC (Event-driven Process Chain) (Keller et al., 1992), Petri Nets (Murata, 1989), etc. Comparison of some of these techniques can be found in (Aguilar-Saven, 2004).

Each of these methods possesses certain advantages and, of course, certain limitations. It is important that the method and underlying technique and tool adequately fit the problem situation in terms of perspective and purpose. Organizational changes often have unpredictable consequences if not properly studied in a controlled manner, before changes are implemented in reality. Modeling and Simulation as a combined methodology, adapted and used for enterprise and organizational (re)design, offers a safe and controlled way to understand how alternative designs and configurations of a system or process would perform (Mastaglio, 1999; Mielke, 1999; Torchiano & Bruno, 2003; Suggs & Lewis, 2007). While modeling provides the static view of the system or process, by visualizing concepts and documenting reality, simulation brings the model to life – by executing the model, showing the dynamic behavior of the system. This includes studying the effects of interactions of (human) actors and reaction to changes.

The existing modeling and simulation theories, methodologies, and approaches allow analysis, design and study of the systems and processes using artifacts (diagrams, notations, languages, tools) specifically designed for this purpose. The experiments and comparison of different alternatives (scenarios) are conducted in a controlled environment. In the past decades, modeling and simulation has gained momentum in the business process field, although at a slow pace, as surveyed in (Melão & Pidd, 2003). Current developments in the simulation field, such as interactive simulation (Robinson, 1994), Web-based simulation (Kuljis & Paul, 2001; Miller et al., 2001), and gaming simulation (Angelides et al., 1999; Mendonça et al., 2006), facilitate experiments that are close to reality, and therefore it makes simulation an attractive practice for the preparation of (re)designing enterprise systems.

Increasingly, simulation is also used to augment analytical models. The two models are used for cross validation as well as for overcoming restrictions in analytical models due to their assumptions that stem from linearity of processes, homogeneity of occurrences, normality, and stationary state of the underlying process.

6 Modeling and Simulation

Simulation techniques have benefited many of the traditional areas in helping to mitigate design flaws, learning about system behavior, providing training, and becoming a standard practice for developing complex systems. Following the analogies of traditional domains, the application of simulation in the context of socio-technical environments such as enterprise modeling, organizational design, and business process (re)engineering, has attracted a huge interest among researchers (e.g., Gladwin & Tumay, 1994; Hlupic & Robinson, 1998; Harrison, 2002; Paul & Seranno, 2003; Vreede, Verbraeck & Eijck, 2003; Seila, 2005). The practice of modeling and simulation is opening a promising research field as the potential and full capacity of enterprise modeling and simulation still has to be revealed (Mielke, 1999; Torchiano & Bruno, 2003; Suggs & Lewis, 2007).

Applying modeling and simulation to an enterprise, and to its organization and business system, could facilitate the understanding of the business domain (e.g., healthcare, banking, commerce) as well as complexity of the extended enterprise (inter- and intra-organizational relationships). Furthermore, it can offer suggestions for significant improvements of the underlying processes (Han et al., 2009). However, more interesting application fields for modeling and simulation emerge with the innovations in linking it to decision-making processes. For example, traditionally, modeling and simulation have been used as instruments to observe the dynamic behavior of systems, to measure IT impacts on organizations, and to study potential outcomes of organizational change. In the current context of open-source software development, service-oriented architectures (SOAs) and business process outsourcing (BPO), modeling and simulation assume an even greater significance in assessing business process management effectiveness, studying the alignment between business process models and corresponding SOAs, and quantifying the alignment between the client and vendor business strategies in a BPO contract. But for these benefits to occur, modeling and simulation ought to be viewed as an *integrated* tool and method. Unfortunately, many popular enterprise modeling methods do not lend themselves for direct simulation (e.g., UML, IDEF, EPC) and the analysts are therefore restricted to merely static modeling. Modeling itself may not reveal sufficient information about the process dynamics and the impact of changes. On the other hand, just using simulation tools may provide little help if there is no profound conceptual modeling preceding it – which is like conducting a complex trip without a roadmap. Simulation without a profound conceptual model would make it difficult, if not impossible, to reach the target, which is to generate accurate and valid output data. Therefore, it is imperative that analysts consider modeling and simulation as an integrated approach in enterprise study, and deem business processes as the main object of design and redesign (Bogusch et al., 1997).

In order to establish enterprise modeling and simulation practice based on profound theories, methodologies, and integrated approaches, the efforts and research in this direction should draw three interrelated theoretical foundations – *the organizational sciences*; the *information systems sciences*; and the principles of *systems engineering* These theoretical foundations complementarily address both the social and technical aspects of modern enterprises. Therefore, many researchers (Greogriadis & Sutcliffe, 2008; Katzensten & Lerch, 2000; Giaglis et al., 1999: Cho et al., 1998) advocate to develop and advance concepts, methods, methodologies, and guidelines for enterprise modeling and simulation. It requires a holistic approach comprising both socio-technical perspectives as well as being based on computational science (Carley, 2002).

7 Why Enterprise Simulation?

Why is enterprise and organizational simulation important? Is it because of change, agility, dynamics, pursuit of marginal profits, quality demand, or are there other drivers for the use of modeling and simulation?

Let us stress two main reasons that explain the relevance and importance of simulation in the enterprise and organizational context – business process driven enterprise information system (EIS) design and dynamics of the business environment. Both trends are characterized by changes and improvements as discussed below. Changes and improvements require experimentation and testing that are at the core of what simulation can do best.

Xu et al. (2007) argue that design and development of future EIS will be increasingly driven by the enterprise business process models. In turn, the role of enterprise business process modeling and simulation as an integral method for system design will be significantly changed as models will be used for code generation and valid models will result in valid systems. Simulation plays a crucial role in redesigning and improving business process models and in the comparison of design options.

Current trends in business process management show (Smith & Fingar, 2003) that processes-oriented approaches are receiving increasing attention in analyzing and designing enterprise business processes. The renewed wave of research interest in business process modeling in general, and in process innovations in particular, is referred to as the *third wave* of business process management (Smith & Fingar, 2003). As the very fabric of process innovation is change, and as changes always need to be evaluated in light of different scenarios and situations, this third wave of business process management demands an even more integral role for modeling and simulation in the design, redesign, and process improvement activities.

Dietz and Hoogervorst (2008) state that due to globalization, removal of trade barriers, and deregulation, the business environment dynamics will rapidly change in the coming years. Future enterprises will have to operate in an even more dynamic and global environment, which requires enterprises to be more agile, adaptive, and transparent. Furthermore, as they affirm, enterprises are purposefully designed systems. This means that we need new skill to design, redesign, and implement an enterprise in a comprehensive and consistent way.

Enterprise level activities require strong decision making support. As stated in (Mastaglio, 1999) enterprise simulation is rather a decision-support environment allowing users to apply, reflect on, and improve their understanding and knowledge about their enterprise. In addition, enterprise simulation is seen as learning and training tool.

Furthermore, continuous competition, increasing capabilities of new technologies and growing customer demands require businesses to keep current and be swift to external changes. Obviously any change is risky and may have serious consequences for enterprises. Early mitigation of risks associated with redesign and innovation is highly desirable, especially in the situations of many uncertainties. Here is where enterprise and organizational modeling and simulation play a significant role to study, analyze, optimize, compare different scenarios, and measure the effects of changes. Application of simulation could be a safe and inexpensive way of studying the impact of changes and revealing the hidden behavior of complex business processes. For simulation to be used in a safe manner, it should be built on high-quality input data and on valid and accurate models. To be inexpensive, it should reuse earlier simulation model parts and be built using existing components as much as possible (Vreede et al., 2003). For example, to deal with complex enterprise models, the models can be hierarchically decomposed into sub-models to ease dealing with them when these submodels are implemented through simulation blocks and organized into libraries for future reuse – like building blocks. Decreasing the complexity of simulation models and increasing reusability of simulation models (blocks) definitely adds to effectiveness and efficiency of enterprise simulation and makes it more cost effective (Park et al., 2008).

8 Summary and Discussion

The aim of this paper was to discuss the relevance, importance and suitability of modeling and simulation in the context of enterprise and organizational study. The paper discussed an enterprise in a more extended sense where its organization and its business processes are taken into account as well. Modern enterprises are hard to study without any relation to the organizational structure they have, to the business processes they consist of, and to the enabling information and communication technologies (ICT). Although the ICT component is not addressed in this paper, it has been discussed that a modern enterprise represents a complex socio-technical phenomenon, where simulation methods can be conveniently applied during their analysis, design, and redesign. But more importantly, simulation can be applied to support decision-making and getting knowledge and understanding of the enterprise. As shown in the literature cited, simulation is already growing into a standard practice in enterprise study and business process management.

Furthermore, the paper attempted to draw a relationship between the notions of enterprise, organization, and business processes and tried to define these notions in relation to each other. The paper also stressed, in part through the discussion of related works, the importance of modeling and simulation as two complementary methods that could transform the current practice of analysis, design, and redesign of enterprises. This emphasis of the enterprise modeling and simulation role may seem especially important and timely in the face of rapidly changing global conditions and environments in which modern enterprises operate – outsourcing, free trade, distributed manufacturing – where enterprise management may wish to study the (dynamic) effects of the change before it has been implemented, or to make other informed decisions.

This article aims at not only re-echoing the relevance and importance of enterprise simulation, but also to become a motivation for a more targeted and extended research program, with discussion and exchange among researchers and practitioners on the tangible values of simulation, and on the practical experiences pertaining to various aspects of enterprise, organization, business processes, and underlying information communication technologies – collectively referred to as socio-technical systems.

References

1. Aguilar-Saven, R.S.: Business process modelling: Review and framework. Int. J. Production Economics 90(2), 129–149 (2004)
2. Angelides, M.C., Paul, R.J.: A methodology for specific, total enterprise, role-playing, intelligent gaming-simulation environment development. Decision Support Systems 25(2), 89–108 (1999)
3. Aris, R.: Mathematical Modelling Techniques. Dover Publications, New York (1995)
4. Arnowitz, J., Arent, M., Berger, N.: Effective Prototyping for Software Makers. Morgan Kaufmann, Elsevier, Inc. (2007)
5. Balci, O.: Guidelines for successful simulation studies. In: Balci, O. (ed.) Proceedings of the 22nd Conference on Winter Simulation, WSC 1990, New Orleans, Louisiana, United States, December 09-12, pp. 25–32. IEEE Press, Piscataway (1990)
6. Barjis, J.: The Importance of Business Process Modelling in Software Systems Design. Science of Computer Programming 71(1), 73–87 (2008)

7. Bogusch, R., Lohmann, B., Marquardt, W.: Process-Centered Modelling and Simulation Environments: Reducing the Cost of Developing Models and Performing Simulation Experiments. EUROSIM - Simulation News Europe 1997(19), 8–10 (1997)
8. Booch, G., Rumbaugh, J., Jacobson, I.: The Unified Modelling Language User Guide. Addison-Wesley, USA (1999)
9. Brandt, D., Hartmann, E., Sander, C., Strina, G.: Designing and simulating sociotechnical systems: Concepts and strategies. Human Factors and Ergonomics in Manufacturing 9(3), 245–252 (1999)
10. Brooks, R.J.: Some Thoughts on Conceptual Modelling: Performance, Complexity and Simplification. In: Proceedings of the 2006 OR Society Simulation Workshop (2006)
11. Brunner, D.T., Cross, G., McGhee, C., Levis, J., Whitney, D.: Toward increased use of simulation in transportation. In: Medeiros, D.J., Watson, E.F., Carson, J.S., Manivannan, M.S. (eds.) Proceedings of the 30th Conference on Winter Simulation, WSC 1998, Washington, D.C., United States, December 13-16, pp. 1169–1176. IEEE Computer Society Press, Los Alamitos (1998)
12. Carley, K.M.: Computational organizational science and organizational engineering. Simulation Modelling Practice and Theory 10(5-7), 253–269 (2002)
13. Carson II, J.S.: Introduction to modeling and simulation. In: Proceedings of the 37th Conference on Winter Simulation, WSC 2005, Orlando, Florida (December 04-07, 2005)
14. Cho, Y.H., Kim, J.K., Kim, S.H.: Role-based approach to business process simulation modeling and analysis. Comput. Ind. Eng. 35(1-2), 343–346 (1998)
15. Dietz, J.L.G., Hoogervorst, J.A.P.: Enterprise Engineering – A Manifesto. In: Dietz, J.L.G., Albani, A., Barjis, J. (eds.) Advances in Enterprise Engineering I. LNBIP, vol. 10, pp. vii-ix. Springer, Heidelberg (2008)
16. Dietz, J.L.G.: On the Nature of Business Rules. In: Dietz, J.L.G., Albani, A., Barjis, J. (eds.) Advances in Enterprise Engineering I. LNBIP, vol. 10, pp. 1–15. Springer, Heidelberg (2008)
17. Gershenfeld, N.: The Nature of Mathematical Modelling. Cambridge University Press, Cambridge (1998)
18. Giaglis, G.M., Paul, R.J., Hlupic, V.: Integrating simulation in organizational design studies. International Journal of Information Management 19(3), 219–236 (1999)
19. Gladwin, B., Tumay, K.: Modeling business processes with simulation tools. In: Manivannan, M.S., Tew, J.D. (eds.) Proceedings of the 26th Conference on Winter Simulation, WSC 1994, Orlando, Florida, United States, December 11-14, pp. 114–121. Society for Computer Simulation International, San Diego (1994)
20. Greogriadis, A., Sutcliffe, A.: A socio-technical approach to business process simulation. Decision Support System 45(4), 1017–1030 (2008)
21. Hancock, W.M., Walter, P.F.: The use of computer simulation to develop hospital systems. SIGSIM Simul. Dig. 10(4), 28–32 (1979)
22. Han, K.H., Kang, J.G., Song, M.: Two-stage process analysis using the process-based performance measurement framework and business process simulation. Expert Syst. Appl. 36(3), 7080–7086 (2009)
23. Harrison, J.R.: Organisational Processes. Simulation Modelling Practice and Theory 10(5-7), 249–251 (2002)
24. Hlupic, V., Robinson, S.: Business process modelling and analysis using discrete-event simulation. In: Medeiros, D.J., Watson, E.F., Carson, J.S., Manivannan, M.S. (eds.) Proceedings of the 30th Conference on Winter Simulation, WSC 1998, pp. 1363–1370. IEEE Computer Society Press, Los Alamitos (1998)
25. IDEF (2008), Family of Methods web page, `http://www.idef.com`

26. Jacobson, I., Booch, G., Rumbaugh, J.: The Unified Software Development Process. Addison-Wesley Longman, Amsterdam (1998)
27. Katzensten, G., Lerch, J.: Beneath the surface of organizational processes: a social representation framework for business process redesign. ACM Transactions on Information Systems 18(4), 383–422 (2000)
28. Keller, G., Nuttgens, M., Scheer, A.W.: Semantische Processmodellierung auf der Grundlage Ereignisgesteuerter Processketten (EPK). In: Veroffentlichungen des Instituts für Wirtschaftsinformatik, Heft 89, University of Saarland, Saarbrücken (1992) (in German)
29. Koehler, J., Hauser, R., Küster, J., Ryndina, K., Vanhatalo, J., Wahler, M.: The role of visual modelling and model transformation in business-driven development. Electronic Notes in Theoretical Computer Science (ENTCS) 211, 5–15 (2008)
30. Kuljis, J., Paul, R.J.: An appraisal of web-based simulation: whither we wander? Simulation Practice and Theory 9(1-2), 37–54 (2001)
31. Larkin, J.H., Simon, H.A.: Why a diagram is (sometimes) worth ten thousand words. Cognitive Science 11(1), 65–100 (1987)
32. Law, A.M., McComas, M.G.: Simulation of manufacturing systems. In: Farrington, P.A., Nembhard, H.B., Sturrock, D.T., Evans, G.W. (eds.) Proceedings of the 31st Conference on Winter Simulation: Simulation—A Bridge To the Future, WSC 1999, Phoenix, Arizona, United States, December 05-08, vol. 1, pp. 56–59. ACM, New York (1999)
33. Lorenzo, O., Diaz, A.: Process gap analysis and modelling in enterprise systems. International Journal of Simulation and Process Modelling 1(3-4), 114–124 (2005)
34. Mastaglio, T.W.: Enterprise simulations: theoretical foundations and a practical perspective. In: Farrington, P.A., Nembhard, H.B., Evans, G.W., Sturrock, D.T. (eds.) Proceedings of the 31st Conference on Winter Simulation: Simulation—A Bridge To the Future, WSC 1999, Phoenix, Arizona, United States, December 05-08, vol. 2, pp. 1485–1489. ACM, New York (1999)
35. Mayer, R.J., Painter, M., de Witte, P.: IDEF Family of Methods for Concurrent Engineering and Business Re-engineering Applications. Knowledge Based Systems, Inc., College Station (1992)
36. Melão, N., Pidd, M.: Use of business process simulation: a survey of practitioners. Journal of the Operational Research Society 54(1), 2–10 (2003)
37. Mendonça, D., Beroggi, G.E.G., van Gent, D., Wallace, W.A.: Designing gaming simulations for the assessment of group decision support systems in emergency response. Safety Science 44(6), 523–535 (2006)
38. Mielke, R.R.: Applications for enterprise simulation. In: Farrington, P.A., Nembhard, H.B., Evans, G.W., Sturrock, D.T. (eds.) Proceedings of the 31st Conference on Winter Simulation: Simulation—A Bridge To the Future, WSC 1999, Phoenix, Arizona, United States, December 05-08, vol. 2, pp. 1490–1495. ACM, New York (1999)
39. Miller, J.A., Fishwick, P.A., Taylor, S.J.E., Benjamin, P., Szymanski, B.: Research and commercial opportunities in Web-Based Simulation. Simulation Practice and Theory 9(1-2), 55–72 (2001)
40. Miller, S., Pegden, D.: Manufacturing simulation: introduction to manufacturing simulation. In: Proceedings of the 32nd Conference on Winter Simulation, WSC 2000, Orlando, Florida, December 10-13, Society for Computer Simulation International, San Diego (2000)
41. Mintzberg, H.: Organization Design: Fashion or Fit? Harvard Business Review 59, 103–116 (1981)
42. Murata, T.: Petri Nets: Properties, Analysis and Applications. Proceedings of the IEEE 77(4), 541–580 (1989)

43. Paul, R.J., Seranno, A.: The process of process reengineering: simulation for business processes and information systems design. In: Proceedings of the 35th Conference on Winter Simulation: Driving Innovation, New Orleans, Louisiana, December 07-10 (2003)
44. Park, S., Kim, H., Kang, D., Bae, D.: Developing a Software Process Simulation Model using SPEM and Analytical Models. International Journal of Simulation and Process Modelling 4(4) (2008)
45. Qureshi, M.N., Kumar, D., Kumar, P.: Modelling the logistics outsourcing relationship variables to enhance shippers' productivity and competitiveness in logistical supply chain. International Journal of Productivity and Performance Management 56(8), 689–714 (2007)
46. Robinson, S.L.: An introduction to visual interactive simulation in business. International Journal of Information Management 14(1), 13–23 (1994)
47. Robinson, S.: Conceptual Modelling for Simulation Part I: Definition and Requirements. Journal of the Operational Research Society 59, 278–290 (2008)
48. Scott, R.W.: Organizations: Rational, Natural and Open Systems, 5th edn. Prentice-Hall, Englewood Cliffs (2002)
49. Seila, A.F.: The case for a standard model description for process simulation. International Journal of Simulation and Process Modelling 1(1-2), 26–34 (2005)
50. Shannon, R.: System simulation: the art and science. Prentice Hall, Englewood Cliffs (1975)
51. Smith, R.: Essential techniques for military modeling and simulation. In: Medeiros, D.J., Watson, E.F., Carson, J.S., Manivannan, M.S. (eds.) Proceedings of the 30th Conference on Winter Simulation, WSC 1998, Washington, D.C., United States, December 13-16, pp. 805–812. IEEE Computer Society Press, Los Alamitos (1998)
52. Smith, H., Fingar, P.: Business Process Management: The Third Wave. Meghan-Kiffer Press, Tampa (2003)
53. Suggs, R., Lewis, B.: Enterprise simulation: a practical application in business planning. In: Proceedings of the 39th Conference on Winter Simulation: 40 Years! the Best Is Yet To Come, WSC 2007, Washington D.C, December 09-12, pp. 205–209. IEEE Press, Piscataway (2007)
54. Torchiano, M., Bruno, G.: Enterprise modelling by means of UML instance models. SIGSOFT Software Engineering Notes 28(2) (2003)
55. van der Aalst, W.M.P.: Trends in Business Process Analysis - From Verification to Process Mining. LNBIP, vol. 12, pp. 27–42. Springer, Heidelberg (2007)
56. de Vreede, G.J., Verbraeck, A., van Eijck, D.T.T.: An Arena Simulation Library for Business Process Modelling. Simulation 79(1), 35–42 (2003)
57. Xu, L., Tan, W.A., Zhen, H., Shen, W.: An approach to enterprise process dynamic modelling supporting enterprise process evolution. Information Systems Frontiers 10(5), 611–624 (2008)
58. Zeigler, B.P., Praehofer, H., Kim, T.G.: Theory of modeling and simulation, 2nd edn. Academic Press, London (2000)

Organization Modeling and Simulation Using BORM Approach

Jiří Brožek[1], Vojtěch Merunka[1], and Iveta Merunková[2]

[1] Czech University of Life Sciences Prague, Faculty of Economics and Management, Department of Information Engineering
{brozekj,merunka}@pef.czu.cz
[2] Czech University of Life Sciences Prague, Faculty of Agrobiology, Food and Natural Sources, Department of Landscape Architecture
merunkova@af.czu.cz

Abstract. This paper presents the BORM approach in the organizational modeling and simulation. The first part of this paper presents the BORM analysis and design methodology, which has proved to be effective in the development of business systems. The second part of this paper shows an example from our regional management project concerning the analysis of the legislation and local officials' knowledge. The project has been related to the processes and agendas of the urban planning of the landscape areas and small settlements with regards to the new housing and building law and regional management trends in the European Union. Our methodology presents necessary aspects of the relevant organizational model without need of deep prior training. Our models were simulated, verified and validated in order to help the officials (especially from the smallest settlements) to improve their knowledge.

Keywords: organization structure modeling, organization structuresimulation, BORM, regional management.

1 Introduction

The Organization Modeling is a vital part of the entire information development process. Darnton [7] and Taylor [28] write, that the major problem (interpreted from viewpoint of the software engineering) with this so-called requirement analysis of organization systems arises in the initial stages of the entire information system development cycle. The initial stage of any methodology today should be concerned with two tasks. The first is the specification of the requirements for the system. The second is the construction of an initial object model, often called an essential object model or conceptual model, built from of a set of domain specific objects known as essential objects. Both these tasks should be carried out with the active participation of the stakeholders, in order to ensure that the correct system is being developed. Consequently, any tool or diagram used at these early stages should be comprehensible to the stakeholders, many of whom are not 'computer system literate'. Moreover, these diagrams must not deform or inadequately simplify requirement information.

J. Barjis (Ed.): EOMAS 2010, LNBIP 63, pp. 27–40, 2010.

The most common technique for requirements specification in current object-oriented methodologies is Use Case modeling as the start of UML documentation process. The concept of use-cases was introduced by Jacobson in the early 1990s [16]. The main information source about UML is the web-site [29]. As Ambler [1] states, Use Cases are often the foundation of most object-oriented development methods. Use Case modeling is concerned with the identification of actors, which are external entities interacting with the software part of the system. This means that in order to employ Use Case modeling, it is necessary for developers to already know the system boundary and distinguish between entities, which are internal and external to that boundary.

It is our experience that the correct identification of the system boundary is a 'non-trivial' task, which often requires significant understanding of the proposed system and consequently can only successfully take place at the end of the requirements specification stage. Some deficiencies in this approach are also highlighted by Barjis in [2]. There are many views on the effectiveness of Use Cases and related tools as a first stage in System Design. Simons and Graham [27] for example describe a situation where Use Case Modeling obscures the true business logic of a system. Because of standard UML-based tools are too oriented at the world of programming concepts, other methods for business logic and process modeling appeared:

1. The basic grammar of other process modeling tools is based on Petri Nets. The strengths of this approach are that it is both graphical and has strong mathematical basis. A practical implementation of Petri Nets is the EPC diagram of Aris methodology [10], for example.
2. Another techniques are based on miscellaneous varieties of flowchart diagrams. This approach is the oldest diagramming technique used in computer science. It was primarily used for visualizing the sequences of operations in computer programs. Today, flowcharts are frequently used to model business processes. A practical implementation of flowcharts is workflow diagram used in Proforma Workbench or FirstStep Business CASE Tools. Indisputably, it is also a kind of the Activity Diagram of UML [12].
3. The third technique used here is the use of state machines. These have the theoretical background ([26] for example), as well as Petri Nets. A practical implementation of state machines is state-chart diagram in UML, for example. Indeed, the sequence diagram of UML has features of state machines as well.

The overview of all approaches for modeling business logic and processes described here is presented in table 1.

2 Our Experience

Our experience in system modeling suggests that classical UML is not suitable for first stages of analysis, where business processes need to be recognized. UML diagrams are too complex for the users from the problems domain community

Table 1. The most used organization modeling approaches

approach	theory behind	advantages	disadvantages
EPC – Aris	Petri Nets	Very popular in Europe, perfectly supported by Aris CASE Tool, easy and comprehensible method for domain experts.	Weak relation at subsequent software development techniques, slow analysis, low expressiveness of large models.
UML Activity Diagram or BPMN	Flowchart	Industry standard, supported by many CASE tools with UML (Unified Modeling Language) or BPMN (Business Process Modeling Notation).	Too software-oriented, difficult to understand by domain experts.
UML sequence and state-chart diagram	Finite state machine	Industry standard, supported by many CASE tools with UML (Unified Modeling Language).	Too software-oriented, difficult to understand by domain experts.
Workflow Diagrams	Flowchart	Easy and comprehensible method for domain experts, perfectly supported by many business CASE Tools.	Weak relation at subsequent software development techniques, not very popular in Europe where Aris takes the dominant place.

as they often contain too much detail concerning potential software implementations. This means classes, inheritance, public/private methods, attributes, link classes, etc. Almost the same experience we have is documented in Simone and Graham [27].

We believe that the business community needs a simple yet expressive tool for process modeling; able to play an equivalent role to that played by Entity-Relation Diagrams, Data-Flows Diagrams or Flow-Charts over the past decades. One of the strengths of these diagrams was that they contained only a limited set of concepts (about 5) and were comprehensible by problem domain experts after few minutes of study. Unfortunately UML approach lost this power.

That is why we developed our own BORM process diagram and our own way to start business system analysis. It is a simple methodology going smoothly from business analysis and simulation to subsequent detailed UML software design based on MDA software-oriented concepts necessary for the construction of software-oriented conceptual model.

2.1 System Development

Developing systems is a complex activity fraught with many difficulties for software engineers as they endeavor to ensure that the right system is built. A right system being one that meets the user's needs at a cost they can afford.

On the surface this would appear a straightforward task, first year university students studying system design are often surprised when it is pointed out to them that incorrectly specifying the required system is one of the major causes of software systems failure. Such students, however, have little experience of the complexity of the real world where software developers and experts from the user domain appear to live in different universes, each with their own jargon, which acts as a barrier to true communication.

It is in this context that software developers face the first and perhaps major challenges of software development; to fully understand the user domain and moreover to convey their understanding of that domain to the user.

Adele Goldberg [14] uses the term "concept space" to describe what the user/experts believe, assumes or knows to be the case. The "articulation space" is what the expert/user communicates in response to the analyst's questions. The analyst then constructs a model to feed back to the user/expert their mental model of the concept space, which they construct out of the information presented in the articulation space. The difference between this analyst's model and the user space is the concept gap.

To a certain extent, part of this gap is unbridgeable; we cannot easily reduce the gap between concept and articulation space as these exist in the user/expert's head. It is true, however, that the languages, natural and graphical, used by the analyst in representing this model, are a vital component in the user/expert's ability to validate this model against the users own concept space.

The problem is to find a common language for the developers to express their understanding of the problem space that is both sufficiently rich for the developers to fully articulate their ideas while also being comprehensible to users from all areas of discourse.

Use-Case has become a well-accepted part of object-oriented analysis and in many cases has proved a useful mechanism for communication between developers and domain experts. We do not intend to discuss it further here. However, Fowler [12] highlights some deficiencies in the Use-Case approach and also suggests that "*Activity diagrams can be useful in cases in which workflow processes are an important part of the users' world.*"

Same as [7], we think that activities are a key component of business process modeling. Eeeles and Sims [9] define a business process consisting of a number of elements; activities, transitions, states and decisions. They state that the UML activity-diagrams can be a useful modeling tool in capturing business processes as well.

Initial analysis diagram should support only problem domain-specific concepts; any software-orientated concepts can be left until later in the modeling process. This is in sharp contrast with UML, which claims to be a universal system; meaning that the same notation is used for analysis, design and documenting the implementation. Our reasons for this requirement are based on the observation that this universality of the UML's notation hinders the design process. In this we are in broad agreement with the criticism of this aspect of UML expressed by Simons and Graham [27].

It is necessary for the organization modeling and subsequent simulation, that every participating object should be viewed as a state machine with states and transitions dependent on the behavior of other objects. Each state is defined by its semantic rule over object data associations and each transition is defined by its behavior, necessary to transform the object from its initial to its terminal state. Organizational and business process models must be able to be simulated. Hence it should accent the mutual relationships (communications and associations) of states and transitions of objects in the modeled system.

3 The BORM Approach

3.1 Motivation

Development of the BORM methodology started in 1993. At that time, several "first generation" object or semi-object-oriented analysis methods (OMT, Martin-Odell, Booch, Coad-Yourdon, Jacobson, etc.) existed. These methods were, and still are, very useful for the development of hybrid software systems. For example an object-oriented client talking to a number of relational servers. However the authors felt that these methodologies possessed two fundamental weaknesses which made them inappropriate for their own development requirements.

Firstly these existing methods did not offer sufficient support for development using a pure object-oriented language like Smalltalk. When developing systems in Smalltalk the authors often used constructs of the language like polymorphism between objects without any inheritance or object dependency, which were not supported and could not be expressed in any of these existing development methodologies. Also in the diagrammatic notations they provided it was impossible to represent most pure object-oriented algorithm. Such algorithms may often be described as mutual asynchronous communications (message passing) between objects, which as the result of receiving messages invoke internal methods with a consequential change in their state.

Secondly, these existing methodologies initially commenced with the construction of a set of classes showing inheritance and aggregation hierarchies. While this is an effective way of expressing the structure required for subsequent coding in an object-oriented language, it is not however effective in illustrating the problem domain. This is because the "object oriented nature" of these diagrams are difficult for domain experts, not educated in computer science concepts, to understand. Consequently such diagrams cannot be used in describing proposed solutions to clients.

3.2 BORM Projects

The initial work on BORM was carried out under the support of the Czech Academic Link Programme (CZALP) of the British Council, as part of the VAPPIENS[1] research project; further development has been carried out with

[1] Visual Application Programming Paradigms for Intergated ENvironmentS.

the support of Deloitte Central Europe. (*VAPPIENS was funded by the British Governments CZALP and administered by the British Council. The authors acknowledge the support they received from this source, which enabled them to meet and carry out the initial work, out of which BORM grew.*) BORM has been used for a number of large projects including

– the identification of business processes in Prague city hospitals,
– the modeling of properties necessary for the general agricultural commodities wholesale sector in the Central European region,
– as a tool for business process reengineering in the electricity supply industry and
– as a tool for business process reengineering for telecommunication network management in the Central European region.

3.3 BORM Fundaments

BORM is a unified approach to business and IT system modeling. For more on the BORM method see [18,20].

BORM is based on the spiral model for the development life cycle as described in [5]. One loop of the object-oriented spiral model contains stages of strategic analysis, initial analysis, advance analysis, initial design, advanced design, implementation and testing.

1. The first three stages are collectively refereed to as the expansion stages. Expansion ends with the finalizing of the detailed analysis conceptual model, which fully describes the solution to the problem from requirements point of view.
2. The remaining stages are called as consolidation stages. They are concerned with the process of developing from "expanded ideas" to a working application. During these the conceptual model is step by step, transformed into a software design.

Object-oriented approach. The object-oriented approach has its origins in the researching of operating systems, graphic user interfaces, and particularly programming languages, taking place in the 1970s. It differs from other software engineering approaches by incorporating non-traditional ways of thinking into the field of informatics. We look at systems by abstracting the real world in the same way as in ontological, philosophical streams. The basic element is an object that describes data structures and their behavior. In most other modeling approaches, data and behavior are described separately, and, to a certain extent, independently. OOP has been and still is explained in many books, but we think that this one [14] written by OOP pioneers belong to the best.

Automata theory. In the field of theoretical informatics, the theory of automata is a study of abstract automatons and the problems they can save. An automaton is a mathematical model for a device that reacts to its surroundings, gets input, and provides output. Automatons can be configured in a way that the output from one of them becomes input for another. An automaton's behavior is defined by a combination of its inner structure and its newly - accepted input. The automata theory is a basis for language and translation theory, and for system behavior descriptions. Its usage for modeling and simulation in software engineering activities has been described in [26] and many newer publications. The idea of automata also inspired behavioral aspects of the UML standard [29].

Three areas of BORM modeling in MDA perspective. MDA (Model-Driven Approach) is a software development methodology. It provides a set of guidelines for the structuring of specifications, which are expressed as step-by-step transformed models. It was created by the Object Management Group (OMG) in 2001 and is the most used software methodology based on the UML (Unified Modeling Language)[29]. BORM can be regarded as a special kind of MDA. In the MDA terminology, we can describe BORM as:

1. The CIM (Computer-Independent Model) modeling, according to the BORM method, is a visualization of the environment in which a project is being executed. It deals primarily with business process models. Its aim is to understand and describe a problem and find a solution. A well-made CIM model enables proper descriptions of settings for information system to be made; a necessary condition for a designed solution. This part of BORM having the special BORM process diagram used for the organizational modeling and simulation is discussed in this paper.
2. PIM (Platform-Independent Model) modeling, according to the BORM method, is a visualization of the required information system in software engineering concepts. The UML (Unified Modeling Language) standard has an important role. There is a set of transforming rules [22] from BORM model to the conceptual UML model [17].
3. The PSM (Platform-Specific) model is a revised form of the PIM model which, unlike PIM, enables specific software implementation, since it includes specific properties of the target environment and reused artifacts of the IT architecture, etc. There is also a set of transforming rules from PIM UML models to the PSM UML models [17].

3.4 BORM CIM – Organizational Modeling

The first part of the method (CIM) covers the organizational modeling. It transforms a project assignment into a model described by miscellaneous hierarchies, process participants, process scenarios, various diagrams and generated reports. The main instrument of verification and validation is the process simulator, which is currently implemented in the Craft.CASE tool [6].

For the following purposes, it is possible to use this part of BORM without any relation to a software engineering phase or organizational structure improvement as is it also presented in the example of this paper. BORM CIM modeling has been used as:

1. Projects documenting processes and organizational structure. These are, for instance, projects whose aim is knowledge management, creating training materials, knowledge visualization, etc.
2. Projects for preparing the groundwork for selection procedures for organizational consultancy, or other consultancy services.
3. Projects for preparing the groundwork for selection procedures for the delivery of information systems, or other software engineering projects.

BORM was initially developed as an object-oriented method for the analysis and design of object-oriented software systems. The process (described by Satzinger [25]) starts from an informal problem specification and provides both methods and techniques, to enable this informal specification to be transformed into an initial set of interacting objects. The tools and techniques developed for requirement analysis and used in the initial phases of BORM, provide an independent method for business process modeling as part of business process reengineering. The authors find that this independent method, referred to as BOBA (BORM Object Behavior Analysis) is frequently used alone.

One advantage of this approach is that it provides a close interactive interchange between the developers and members of the user's organization. As well as identifying initial objects, BOBA elicits from the domain experts, detailed descriptions of their requirements which are fed back to them via easily understood descriptions of the proposed system's behavior using a number of tables and graphs.

The problem specifications from which the process starts are obtained from relevant parties in the problem domain by interviewing. This determines a list of required system functions, which are essentially Use Cases.

From this list, a set of system scenarios is formed. BOBA scripts always include at least the four sections shown in Table 2.

Table 2. Scenario structure in BORM

section name	description
initiator	A brief verbal description of the beginning of the scenario including any inputs or entry conditions. It also describes the first event or first activity of some element within the process.
action	A verbal description of the process itself.
participants	The set of those members of the system, which are required for the action. It is often the case that the same participants may be present in several processes of the modeled system.
result	A brief verbal description of the end and outputs of the scenario.

This structure from table 2 represent the four most important attributes of each scenario. The complete set of scenarios is capable of describing system behaviors, as well as determining the objects that perform these behaviors. In addition to those four attributes each scenario must also refer to the required system function it realizes.

3.5 BORM Business Diagram

BORM uses an original diagram for business process modeling and subsequent simulation (see figure 1). It conveys together information from three separate UML diagrams: *state*, *communication* and *sequence*. The BORM group has found that it is clearly understood by business stakeholders. Main principles of the BORM process diagram are:

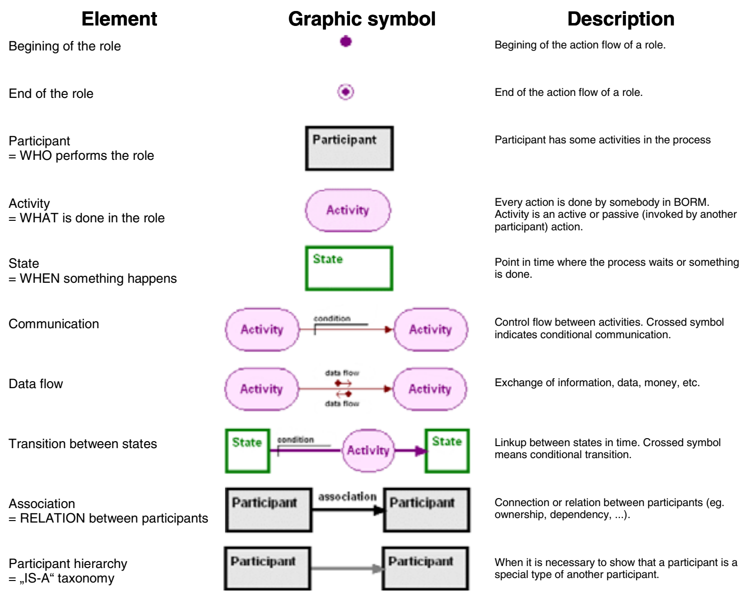


Fig. 1. BORM diagram symbols

1. Each subject participating in a process is displayed in its states and transitions.
2. This diagram expresses all the possible process interactions between process participants. The business process itself consists of a sequence of particular communications and data flows among participating subjects.

More formally, BORM process diagrams are graphical representations of interconnected Mealy-type finite state machines of particular subjects. The idea of modeling objects as finite-state machines was firstly discussed in [26]. Visual simulation of a business process is based on market-graph Petri net. This similar approach is described in detail by [3]. Therefore we can show states, transitions and operations for all subjects playing a role in a business process. This is a very powerful, yet simple diagram.

4 BORM Application Example –Public Regional Management System

One of the recent BORM applications of organizational modeling and simulation was the project of improvement the decision-making on the level of mayors and local administrations. It offers the possibility to model and simulate real life situations in small settlements. The project activities were for modeling, simulation and reengineering processes related to the regional government processes of small towns and villages, and the subsequent development of supporting information systems addressing life situations of local people.

Nowadays we have to solve many problems related to the small settlement development and expansion, landscape care and over-all efforts to improve the quality of life and the level of democracy while preserving the conditions of the sustainable development (addressing living standard, cultural and historic value, agricultural and industrial production, transport infrastructure construction, tourism potential, etc.).

One of the specific problems that our approach can be applied to is the *urban sprawl* as it is stressed by Frumklin in [13]. The cause of the *urban sprawl* in the small settlement development is the fact that the elected members of local administrations (e.g. mayors and clerks) are not (and as the logic states they cannot be) fully educated in all the details of law, state and local administration agenda and their effects on living in the settlements. They don't know how to use fully the legislation in favor of the settlements and usually depend on a misleading interpretation provided by their governing bodies and more often by another subjects (usually privately involved in the process in question and thus biased).

Urban sprawl is a phenomenon that emerged in the last decades in the advanced industrial countries (USA, France, Great Britain) and recently also in our country. Inhabitants of affected settlements usually percieve the urban sprawl positively at first, mainly because of the lobbying. It can be described as an uncontrolled expansion of certain kind of urban build-up into the free landscape caused by favorable land prices, demand for cheap but modern estates, etc. Dualny and others write [8] about harmful absorption of original small settlement structures, which causes following negative effects:

1. Pawning of *infrastructure development* of the original settlement. New inhabitants fulfil themselves and shop only at the place of their work in a metropolis and the settlements are just a kind of sleeping accomodation for them. New inhabitants' lack of interest in contributing to the settlement

development leads to misusing of democratic principles of the self administration against the original local inhabitants and inevitably to the rise of social segregation between the original and the new inhabitants.
2. *Urban sprawl* causes disruption of the cultural and historical value of the settlement, disruption of the ecological stability of the area, deconstruction of the transport infrastructure, loss of touristic attractiveness etc.
3. *Loss* of the quality agricultural soil.

4.1 Modeling and Simulation

We analyzed the legislation and local officials' knowledge related to the processes and agendas of the urban planning of the landscape areas and small settlements with regards to the new housing and building law and regional management trends in the European Union.

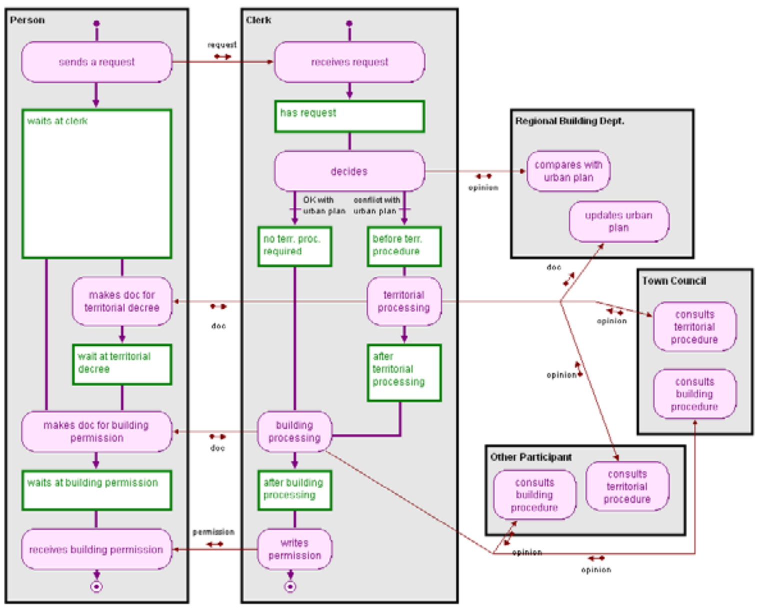


Fig. 2. Building permission process

Our approach using process models and their visual simulation helps the officials (especially in the smallest settlements) to clarify the legislation and shows them possible ways of its usage. Our models and their visual simulation show how the BORM can be used to improve the process of decision-making on the level of mayors and local administrations. It offers the possibility to model and simulate real life situations in small settlements. The example at the figure 2 shows the BORM business object diagram of a process of obtaining building permission. The figure 3 shows the concrete simulation step.

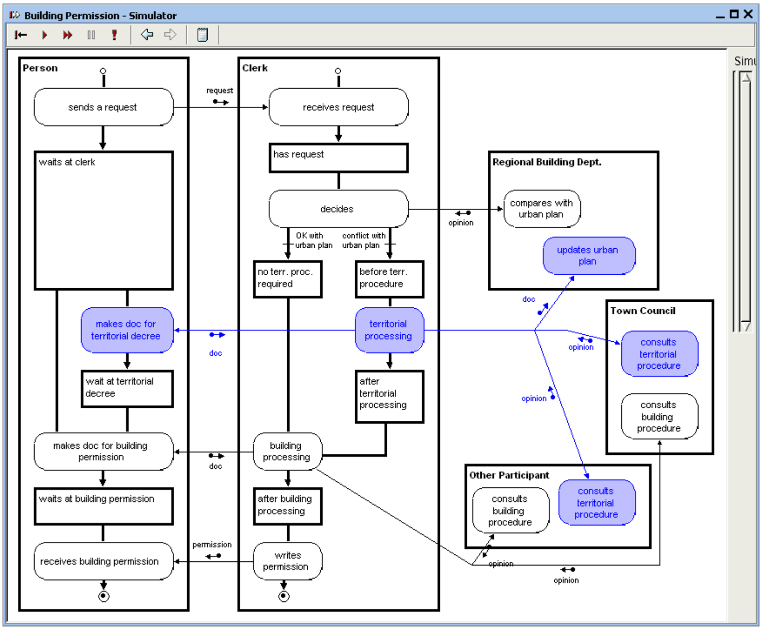


Fig. 3. Simulation step example in Craft.CASE tool [6]

5 Conclusion

BORM is an object-oriented and process-based analysis and design methodology, which is proven to be effective in the development of business systems. The effectiveness gained is largely due to an unified and simple method for presenting necessary aspects of the relevant business model, which can be simulated, verified and validated for subsequent software implementation. Moreover, several partners of our projects use miscellaneous legacy Process Modeling Systems for historical reasons (e.g. EPC-based ARIS, for example). However they prefer to analyze and design processes using BORM as well. Later they convert the results into their legacy systems.

We feel that the highest value of BORM is generated by the way of modeling, which covers two different worlds: business engineering and software engineering. BORM is a comprehensible tool for the collaboration between system architects and problem domain experts via organization structures modeling and subsequent simulation.

References

1. Ambler, S.: Building Object Applications That Work, Your Step-By-Step Handbook for Developing Robust Systems Using Object Technology. Cambridge University Press & SIGS Books (1997), ISBN 0521648262

2. Barjis, J.: Developing Executable Models of Business Systems. In: Proceedings of the ICEIS - International Conference on Enterprise Information Systems, pp. 5–13. INSTICC Press (2007)
3. Barjis, J., Reichgelt, H.: A Petri Net Based Methodology for Business Process Modeling and Simulation. In: Proceedings of the Fourth International Workshop on Modeling, Simulation, Verification and Validation of Enterprise Information Systems (MSVVEIS), Paphos Cyprus (2006), ISBN 972-8865-49-X
4. Bellin, D., Simone, S.S.: The CRC Card Book. Addison-Wesley, Reading (1997), ISBN 0-201-89535-8
5. Boehm, B.W.: Software Engineering Economics. Prentice-Hall, Englewood Cliffs (1981)
6. Craft.CASE analytical and modeling tool, Craft.CASE Ltd. Torriano Avenue London, http://www.craftcase.com
7. Darnton, G., Darnton, M.: Business Process Analysis. International Thomson Publishing (1997), ISBN 1-861-52039-5
8. Dualny, A.: Suburban Nation: The Rise of Sprawl and the Decline of the American Dream. North Point Press (2001), ISBN 978-0865476066
9. Eeles, P., Sims, O.: Building Business Objects. John Wiley & Sons, Inc., New York (1998)
10. EPC- Event-driven Process Chain, http://en.wikipedia.org/wiki/Event-driven_process_chain
11. Eriksson, H., Penker, M.: Business Modeling with UML. John Wiley and Sons, Chichester (2000), ISBN 0-471-29551-5
12. Fowler, M.: UML Distilled: Applying the Standard Object Modeling Language. Addison Wesley, Reading (1997)
13. Frumkin, H., et al.: Urban Sprawl and Public Health: Designing, Planning, and Building for Healthy Communities. Island Press (2004), ISBN 978-1559633055
14. Goldberg, A., Rubin, K.: Succeeding with Objects – Decision Frameworks for Project Management. Addison-Wesley, Reading (1995), ISBN 0-201-62878-3
15. Hall, J.A., et al.: Accounting Information Systems. Cengage Learning EMEA (2004), ISBN 0324560931
16. Jacobson, I., et al.: Object– Oriented Software Engineering – A Use Case Driven Approach. Addison-Wesley, Reading (1991), ISBN 0-201-54435-0
17. Knott, R.P., Merunka, V., Polak, J.: Process Modeling for Object Oriented Analysis using BORM Object Behavioral Analysis. In: Proceedings of Fourth International Conference on Requirements Engineering, ICRE 2000, Chicago, USA. IEEE Computer Society Press, Los Alamitos (2000)
18. Knott, R.P., Merunka, V., Polak, J.: The BORM methodology: a third-generation fully object-oriented methodology. In: Knowledge-Based Systems. Elsevier Science International, New York (2003), ISSN 0950-7051
19. Kotonya, G., Sommerville, I.: Requirements Engineering: Processes and Techniques. John Wiley and Sons, Chichester (1999), ISBN 978-0-471-97208-2
20. Liu, L., Roussev, B., et al.: Management of the Object-Oriented Development Process - Part 15: BORM Methodology. Idea Publishing, USA (2006), ISBN 1-59140-605-6
21. MDA – The Model Driven Architecture, OMG – The Object Management Group, http://www.omg.org
22. Merunka, V., Brozek, J., Nouza, O.: Automated Model Transformations Using the C.C Language. In: EOMAS 2008, Montpellier, France. LNBIP, vol. 10, pp. 137–151. Springer, Heidelberg (2008)

23. MetaCase - Domain-Specific Modeling with MetaEdit+, http://www.metacase.com
24. Partridge, C.: Business Objects - Reengineering for Reuse. Butterworth-Heinemann, Butterworths (1996), ISBN 0-7506-2082-X
25. Satzinger, J.W., Orvik, T.U.: The Object-Oriented Approach - Concepts, Modeling and System Development. Boyd&Fraser (1996)
26. Shlaer, S., Mellor, S.: Object Lifecycles: Modeling the World in States. Yourdon Press (1992), ISBN 0136299407
27. Simons, A.J.H., Graham, I.: 30 Things that go wrong in Object Modelling with UML 1.3. In: Kilov, H., Rumpe, B., Simmonds, I. (eds.) Behavioral Specifications of Businesses and Systems, pp. 237–257. Kluwer Academic Publishers, Dordrecht (1999)
28. Taylor, D.A.: Business Engineering with Object Technology. John Wiley, Chichester (1995), ISBN 0-471-04521-7
29. The UML standard, OMG – The Object Management Group ISO/IEC 19501, http://www.omg.org
30. Yourdon, E.: Mainstream Objects – An Analysis and Design Approach for Business. Prentice-Hall, Englewood Cliffs (1996), ISBN 0-13-209156-9

Agent-Based Simulation for Evaluating Flexible and Agile Business Processes: Separating Knowledge Rules, Process Rules and Information Resources

Yiwei Gong and Marijn Janssen

Faculty of Technology, Policy and Management, Delft University of Technology, Jaffalaan 5, 2628 BX Delft, The Netherlands
{Y.Gong,M.F.W.H.A.Janssen}@tudelft.nl

Abstract. In today's ever changing environment organizations need flexibility and agility to be able to deal with changes. Flexibility is necessary to adapt to changes in their environment, whilst agility is needed to rapidly response to changing customer demands. In this paper a mechanism based on the separation of knowledge rules, process rules and information resources is proposed to enhance the flexibility and agility of business processes. Agent-based simulation is employed to test and evaluate if the necessary flexibility and agility can be created using these mechanisms. In the agent-based simulation the clients and organization entities are simulated, whilst the separation of knowledge rules, process rules and information sources is implemented. Three changes have been evaluated using the simulation. Flexibility and agility proved to be dependent on good system design based on the separation of knowledge rules, process rules and information sources, as well as on the human resources and capabilities executing tasks. Agent-based simulation proved to be a suitable tool for evaluating the level of flexibility and agility prior to real implementation.

Keywords: Agent-based Simulation, Flexibility, Agility, Business Process Management.

1 Introduction

The creation of flexible business processes has received more and more attention by organizations to remain competitive, to satisfy customer wishes and to be able to react to the competitive environment [1]. A business process is the time-dependent sequence of activities and the coordination of these activities is often supported by workflow management (WFM) [2] or business process management (BPM) systems [3]. Flexibility is the general ability to react to changes [4], whilst agility is the speed in responding to variety and changes [5]. Flexibility and agility have different emphasises [6]. *Flexibility* is viewed as the various dimensions influencing change in an organization. For example, governmental organizations have to implement the legal requirements that laws and regulations pose on the tasks that are performed by the organizations and to what extent these requirements are actually met. As law and regulations are subject to frequent change, it will require a major effort of organizations, and their supporting IT applications. Organizations need to make sure their business services and processes

J. Barjis (Ed.): EOMAS 2010, LNBIP 63, pp. 41–58, 2010.

and supporting applications are *flexible* enough to adapt to changing situations. In contrast, *agility* focuses on the speed in dealing with changes from a customer point of view. For example, an insurance company is going to develop a new insurance product (or service) in a timely manner to take up the market, and a new business process has to be created quickly to support the new product.

Traditional WFM and BPM systems use business process models as a kind of scheme to define the 'recipe' of a process execution [7]. Such an imperative modelling implies that the system will actually and strictly carry out tasks in a pre-defined sequence as determined by the business model. Every time a change happens, modelling experts have to carefully adapt the related models and validate them. Even for a minor change, adapting the process model is unavoidable and the full process management cycle which might include various validations and tests must be followed. The efficiency of the adaptation depends on both the employed modelling technology and the skills of the experts. Heinl [8] provides a list of problems with traditional BPM approaches:

- *It is almost impossible to identify all process steps a priori.* This issue is caused by the enormous complexity of the process.
- *Even if a step is identified, it is not obvious whether it should be included into a certain process.* To model all alternative paths at all relevant points of the process would blow up the size of process models and decrease their readability drastically.
- *It is not always possible or desired to prescribe the sequence of control and correction steps in detail.* Steps can be used in numerous processes. It totally depends on the specific situation, which steps have to be executed in which order.

Taking the lessons from traditional BPM systems, we belief that in a complex BPM system, flexible business processes should avoid neither to be too rigid nor to impose strictly predefined execution procedures. In this paper we propose the separation of knowledge rules, process rules and information resources as a mechanism to enhance the flexibility and agility of business processes.

Agent-based simulation is currently in widespread use to catch the behaviours of complex systems in which multiple entities are involved [9]. An agent is anything that can be viewed as perceiving its environment through sensors and acting upon that environment through actuators. Specially, in artificial intelligence, an intelligent agent is an entity which can observe and act upon an environment and directs its activity towards achieving goals [10]. Generally speaking, typical/classical agent-based simulations tend to model (relatively simple) behaviours of individuals and present the "emergence" (natural or social) phenomena of the whole. The system behaviour is made up by the interactions among agents. Agent-based simulation has been used in the study of supply chain process for a long time [9]. Agent-based simulation is also employed to evaluate the design and strategy of organizational structure [11]. Yet agent-based simulation for the creation of flexible and agile business processes has not received much attention.

AGILE is an acronym for Advanced Governance of Information services through Legal Engineering. Within this project we aim at developing a design method, distributed service architecture and supporting tools that enable organizations - administrative and otherwise - to orchestrate their legal information services in a networked environment. The Dutch Immigration and Naturalisation Service (Immigratie en Naturalisatiedienst, IND) is the partner that provides the empirical setting for our experiments, which is subject to continuously changes in the legislative environment requiring them to change their business processes and therefore flexibility is needed. Agility is of importance as the number of persons applying for a permit to stay in the Netherlands fluctuated significantly over the years.

In this paper we will start explaining the ideas and concepts for creating flexible and agile BPM. These ideas are incorporated in an agent-based simulation to ensure that both clients and organizational entities are modelled. The agent-based simulation consists thus of simulated agents and emulated mechanisms for improving the adaptability and flexibility. Rather than modelling norms used by intelligent agents we use agent-based simulation to experiment with dynamic processes of agent interactions in the context of a concrete design. This is to understand the impact on the system of introducing mechanisms to create flexibility and agility. In the next section we will introduce the key areas of our approach. In Section 3 we will explain how techniques from each area will be combined in order to deal with the issues described above. We will then illustrate them by some examples and a use case in Section 4. We will conclude this paper by drawing conclusions and by discussing our future research plans.

2 Background

This section will explain the key research areas that all contribute to our combined approach on how to ensure flexibility and agility, given changing legal demands and constraints.

2.1 Flexibility and Agility

The word '*flexibility*' can be defined in several different ways depending on the discipline or the nature of the research [12]. In the information system domain, flexibility is the ability of organizations to respond to changes in the environment [4]. Flexibility is a multidimensional concept and various conceptualizations can be found in the literature [13-15]. These dimensions include robustness, modifiability, new capability, and partnering flexibility [15]. The first dimension of flexibility is robustness, the ability to endure variations and perturbations, withstand pressure, or tolerate external changes. This relates to situations in which an organization has the built-in capacity to address uncertainty for varying levels of demand, product mix, and resource availability [13]. The second dimension, modifiability, is the ability of an organization to make modifications (e.g., to adjust existing product attributes or alter service composition) to cope with less foreseeable events when they occur. An example is the altering of existing business rules without major setup efforts. New capability is the third dimension referring to the ability to innovate in response to dramatic changes or novel situations. As a network might change and new services might be added and removed. Service flexibility is included as the fourth dimensions, which is similar to partnering flexibility [14]. This is the ability to allow the transfer of services to other organizations or to be brought in house. In summary, flexibility is the ability of organizations to react to changes and covers various dimensions.

The concepts of '*agility'* is widely collected and deeply discussed by various authors. Wadhaw [6] and Bernardes [16] provide overviews, and both conclude that there is no consensus on the definition of agility. 'Speed' as a important perception of agility was indicated by [6]. In this paper we adhere that agility reflects the speed in dealing with changes. Typical definitions reflecting the speed aspect include:

- "*The ability to* ***accelerate*** *the activities on a critical path that commences with the identification of a market need and terminates with the delivery of a customized product*" [17]
- "*The ability to produce and market successfully a broad range of low cost, high-quality products* ***with short lead times*** *in varying lot sizes, which provide enhanced value to individual customers through customization*" [18]
- "*The ability of an enterprise to respond* ***quickly*** *and successfully to change*" [5]
- "*The capability of surviving by reacting* ***quickly*** *and effectively to changing markets, driven by customer-designed products and services*"[19]
- "*The organization's capacity to gain competitive advantage by intelligently,* ***rapidly*** *and proactively seizing opportunities and reacting to threats*"[20]

Those definitions are concluded as sufficiently representative in the understanding of the concept of agility [16]. Although those definitions are still more or less concerning some abilities, which are overlapped with the concept of flexibility, highlighting the 'speed' aspect can emphasize the ability that is not emphasized in flexibility and gives prominence to this aspect. Applying the speed perception in business processes, agility is the ability to provide quick response to the changes.

Flexibility and agility actually present different perspectives in responding the changes in environment. Flexibility is the *organization* perspective and covers multiple dimensions including the capability of transforming business processes and the diversity of business processes that can be achieved. In contrast agility can be viewed as a single dimension: time, which does not concern the organization, but primarily the *relationship with the clients*. In dealing with changes from a dynamic and complex environment, both of them are necessary. Flexibility is needed to react to changes in the environment (legislation in our case study) and agility is needed to react quickly to customers.

2.2 Creating Flexibility and Agility BPM

BPM includes methods, techniques, and tools to support the design, enactment, management, and analysis of operational business processes. It can be considered as an extension of classical Workflow Management (WFM) systems and approaches [3]. WFM is process orientated, which refers to the time-ordered sequence of activities. A workflow shows the sequence of activities to be performed and how a particular case, normally an application, should be dealt with. Besides stating how information is operationally processed, BPM also involves the consideration of what information is offered from whom to whom. Such an extension refers to a service perspective and brings a possibility to align the current popular Service Oriented Architecture (SOA) thought way. SOA is "a framework for integrating business processes and supporting IT infrastructure as secure, standardized components—service—that can be reused and combined to address changing business priorities" [21].

BPM has involved flexibility and agility issues for a long time. Some contemporary research like [22] and [2] provide a clear description on how static workflow was implemented in the first half of 1990s. The second generation of BPM systems had already taken flexibility issues into account. [23-24] elaborated on the consideration of "separation of process and rule". Müller [25] summarized this as a main characteristic in BPM in the first half of 2000s. A process defines ways for the organization to interact with its internal entities (e.g. applications, staff, and departments) and with its external entities (e.g. partners, customers). It is about how to conduct the business organizations. A *rule* represents the logic and knowledge in decision making. It is about what rather than how [23]. The consideration of this separation is that in many knowledge-intensive organizations, large sets of rules are involved. Rules may change frequently and should be managed by people other than the ones executing business process. This is also called "separation of concerns" [24]. To be precise and avoid confusion with the concept of business rules, we employ the words '*knowledge rules'* to denote the logic and knowledge in decision making. Knowledge rules are necessary for making decisions, whereas, business rules are focused on which tasks are subsequently needed to be triggered.

This solution does bring more flexibility and agility than the traditional way did, but it still does not address several crucial issues, such as multiple processes initiating and integrated services delivering [1]. Approaches based on the concept of Event-Condition-Action (ECA) are proposed to achieve a higher flexibility and agility [1]. Instead of using business rules to describe pre-defined business processes, ECA gives guidelines on how to construct a process. Here, we use the word 'process rules' to distinguish it from business rules. A '*process rule'* provides a specification on building a specific process for a given event. The building blocks (resources), such as web services, are managed separately. The execution of building such a process is carried out by a role which we label 'orchestrator'. Therefore, in order to have the maximum flexibility in a BPM system, we propose that the management of knowledge rules, process rules and information resources needs to be separated. This principle forms the basis for creating flexibility and agility. The separation of these parts will be implemented in an agent-based simulation and the simulation is employed to evaluate the level of flexibility it creates.

2.3 Agent-Based Simulation

Simulation is an experimental approach in scientific research to validate hypotheses and explore scientific findings [26]. In a simulation computers are used to evaluate a model numerically, and data are gathered in order to estimate the desired characteristics of the model [26]. The term *agent*, or *software agent*, has been widely used and has found its way into a number of technologies. In artificial intelligence, an agent is an entity which can observe and act upon an environment and directs its activity towards achieving goals [10]. For simulation purposes agents can be used to model an organizational entity which can include individuals, departments or even organizations. Agent-based simulation refers to an experiment in which the dynamic processes of agent interactions are simulated repeatedly over time, as in systems dynamics, and time-stepped, discrete-event simulation. It is expected that simulations can demonstrate our design concept and design compliance. Three notions of agents are important in the agent-based simulation.

Agents as Service Providers and Consumers. Agents are used to represent human beings and/or organizational entities. A process is created by letting these entities conduct certain tasks. The use of a recently popular style, Service Oriented Architectures (SOAs), has shifted modelling attempts from process orientation to service orientation [27]. Processes can be created by invoking a sequence of services. Reorganizing business as services embodies current activities of business innovation. By requesting information and responding to each other, agents act the role of service providers and consumers. In a process, an agent can be a service consumer on the current step, but a service provider in the next step.

Although a simulation environment is a simplification of a practical environment and an agent might not reflect all the requirements of a service (e.g. missing controls on security aspect), the evaluation of service design and service combination on functions using agent-based simulation is feasible.

Agents as Software. Essentially, agents are software applications, but run in a simulated environment. The functions of targeted applications can be presented by agents. Therefore in the simulation the agents as software can be regarded as a prototype of the future system (i.e. doing the real thing instead of abstracting it from reality). Furthermore, it provides a test bed to evaluate the functionality of a future system, as well as the function decomposition and deployment on software components of the future system, and thus reduces the risk of extra system development in the future.

Agents as orchestrators. The various agents need to be orchestrated, i.e. they need to be aware of each other, and they need to provide each other information and so on. Orchestrators are boundary spanners and ensure that parts are related to each other. This is similar to ‘linking pins’ in organizational science [28]. At least a directory of agents is necessary for agents in order to locate each other. Although it is outside the scope of this paper, the agent-based simulation can be used to test, evaluate and compare various orchestration mechanisms. This can help us to decide on the orchestration solution in the future system and improve the chances of successful system development.

These three notions of agents make up the core of the agent-simulation environment, which architecture and implementation will be discussed next.

3 Agent-Based Simulation Environment

3.1 Agent-Based Simulation Architecture

Based on the discussion above, we developed a simulation architecture whose structure is shown in Fig. 1. The architecture consists of client agents who represent customers in need for a certain service, processor agents who are able to fulfil a request of a client by providing services and one orchestrator agent who matches the client and the processor or the processor and a manager agent. Processor agents use the knowledge from the knowledge layer to fulfil their tasks.

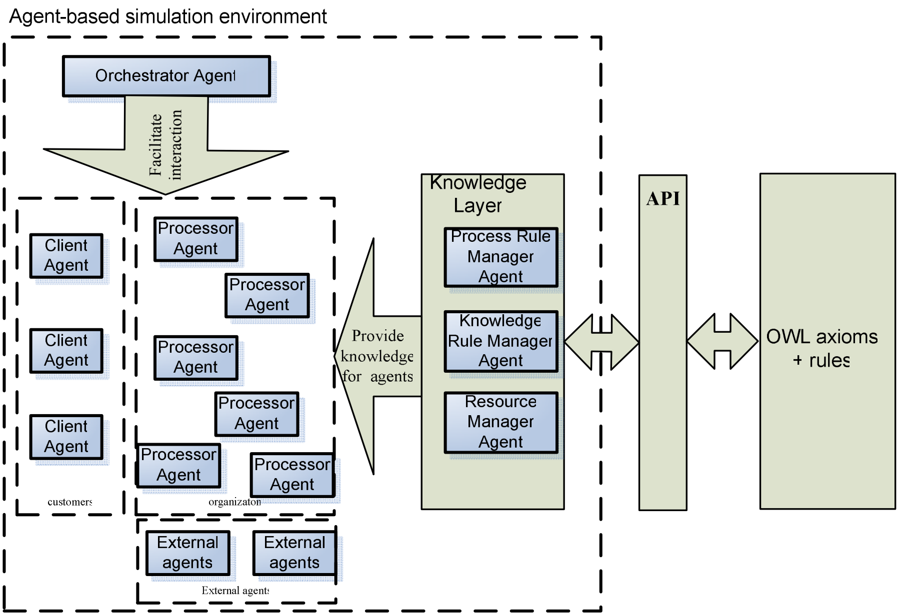


Fig. 1. Agent-based Simulation Architecture

Client Agent: Client agents represent persons submitting applications. In the implementation the client data (e.g. personal information) will be stored in the client agent and used as an input for an application. The number of client agents represents the number of people applying for a residence permit in our case study.

Processor Agent: Processor agents represent the internal business processes of the organization. Each state of a processor agent refers to a next step in the process. The state of an agent will be changed only after a certain task is started by sending out or receiving a message. During the processing for a certain application, a client agent will converse with only one certain processor agent who acts as the case handler in this way ensuring a single point of contact. A processor agent can invoke other processor agents to commission a sub-process. Therefore a processor agent will be dominated either by a client agent or another processor agent. That means a processor agent will be unavailable until the process triggered by a client agent or a sub-process assigned by another processor agent is finished. The number of processors represents the processing capability of the organization.

***External agent*:** External agents represent services that are outside the organizations. In the example below the service to access a database that contains information on bachelor and master degrees obtained at Dutch higher educational institutions is used.

Orchestrator Agent: The Orchestrator creates business processes by managing the sequence of agents, i.e. the orchestrator determines the agents that will execute the next step based on a number of factors. Which processor agent will be assigned to a client agent is decided by the orchestrator agent. It will match a client agent with a processor agent based on its awareness on the availability and utilization of processor agents. The orchestrator has an overview of all processor agents and its attributes.

Knowledge-based Agent: Knowledge-based agents enable the reasoning on rules/-models of rules. They should have their own rule repository or be connected via some API to Ontology Web Language (OWL) model and other components (maybe from other platforms) which can provide reasoning services. The *Knowledge Rule Manager* agent contains knowledge and criteria in decision making, which decides whether the application will be granted. The *Process Rule Manager* agent involves models or rules on processes building. Process rules indicate how the system should deal with a given application based on the situation of input application and the state of the system. The *Resource Manager* provides information of various information/data resources. It will record and inform other agents about the providers of certain information. Those providers can be legacy systems, web service providers, local databases or even a staff in a certain department.

OWL Component: The OWL axioms and rules are provided by a component independent of the simulation environment. This external component provides necessary knowledge input to the knowledge-based agents. The knowledge engineering itself is a complicated approach and therefore out of the scope of this paper. As this part is not simulated, potentially a real knowledge application can be used.

API: The API facilitates the connection between knowledge-based agents and the OWL component. Involving the API is on the one hand necessary for ensuring the independence of the agent simulation environment and the OWL modelling environment, and on the other hand easier for interaction.

In the architecture, the concepts of "agents as service providers and consumers"", "agents as software" and "agents as orchestrators" are used. First of all, the Processor agent works as a service provider for the Client agent, as the only one agent that keeps a conversation with the Client agent. All the other complex back-end processes are hidden behind the Processor agent. A similar relationship can be identified between the Processor and any of the manager agents in which the complexity is encapsulated. In this way the functionality of agents can be built, added, tested and selected one by one. Once a single agent properly functions, it can be included to the simulation to test whether the behaviour is correct within a multi-agent setting and if the data format and structure enables the interaction with other agents. They can be trusted and applied in a real software development project. For example, we can build a model in the Knowledge Rule Manager agent to compute the decision result of highly skilled migrant applications (an elaborated use case can be found in next section). If it is successful, then the algorithm, such as condition-action rules composition, and the structure of data, such as a decision tree or a decision table, can be reused in a software application used in the real world. Finally, optimization agents communication, by means of orchestration, can be tested before it is applied in a real environment. In other words, if we find out that all the agent communications are being directed by the Orchestrator agent can result in a higher efficiency and robustness, then we can decide to adopt this mechanism in a real solution.

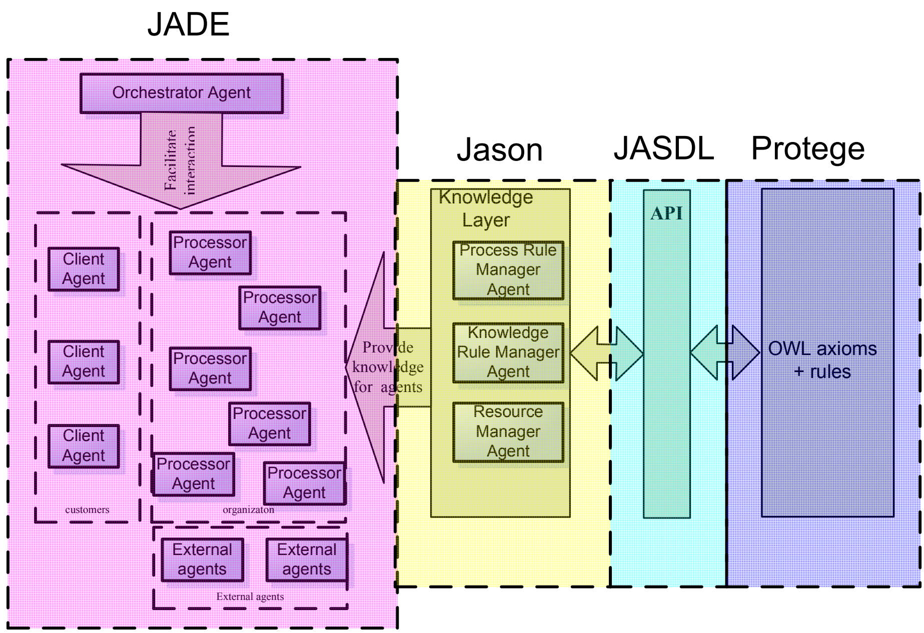


Fig. 2. The Implementation of the Agent-based Simulation Architecture

3.2 Implementing the Agent-Based Simulation

The implementation of the architecture was realized by employing several software components, which are selected based on their functionalities and level of openness (i.e. the use of open standards and open source software). The mapping of the software on the architecture is visualized in the figure below. The components are explained in more detail in this section.

The agent-based simulation is implemented in JADE (Java Agent DEvelopment Framework) environment. Using a computer system to prototype and manage experiment processes has been realized as a crucial support for scientists; and JADE is an appropriate tool for prototyping as it provides feasibility for enabling semantic information between distributed components and freedom to realize complex interactions [29]. JADE is also useful for creating and deploying a distributed agent organisation modelling workflow model and for system monitoring at the level of agents and communications [30]. It should be mentioned that all the JADE components in Fig. 2 are the custom-made components for the simulation. JADE provided components (infrastructure) are not included. The useful infrastructure provided by JADE includes: a graphical user interface (GUI), a DF service (Directory Facilitator, i.e. a yellow page agent), a sniffer which can present a sequence diagram of message communication. Fig. 3 is a screenshot of the sniffer agent GUI and provides an example of the agent interaction in JADE environment. This example is based on a simple scenario in which one client agent (applicant1), one processor agent (processor1) and a knowledge rule manager agent (KRM) interact with each other under the instruction of an orchestrator agent (INDOrchestrator). The orchestration is facilitated by a DF agent (df) provide by JADE.

JADE is actually a Java framework. That means the executable rules (e.g. knowledge rules or process rules in foregoing discussion) implemented in JADE environment is in form of Java code which is imperative. As what we discussed in the introduction chapter, imperative modelling is not able to bring the desired flexibility and agility. Therefore a tool which can represent the rules in a more formal way is needed. This requirement brings us to the consideration of Jason.

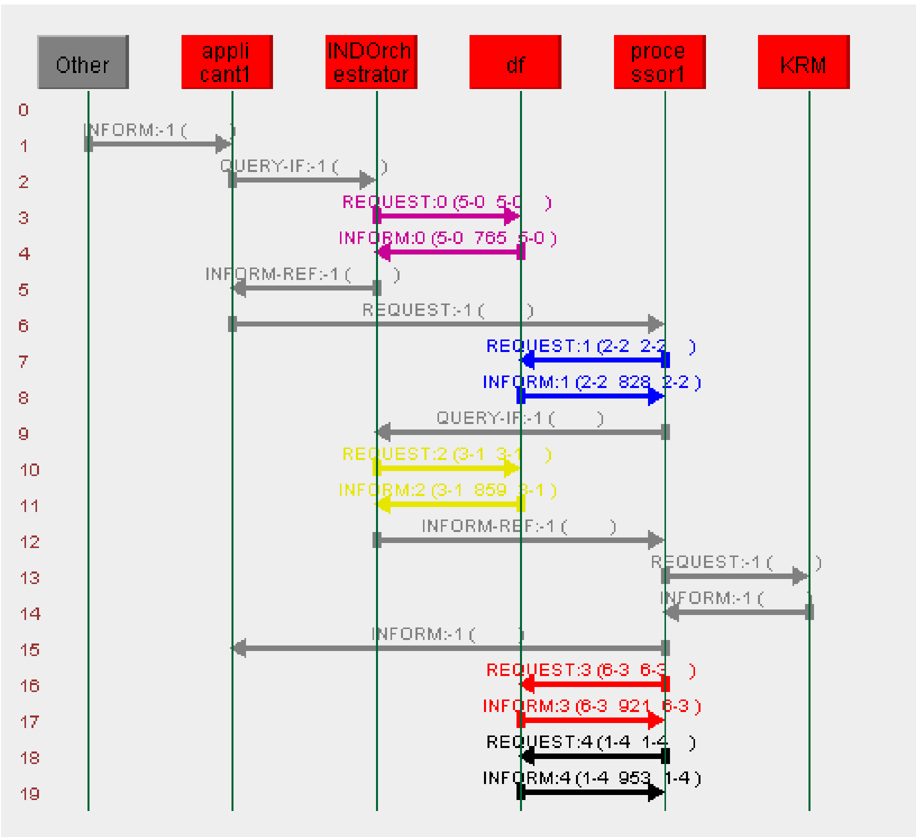


Fig. 3. Example Sequence Diagram of Agent Interactions

Jason is an interpreter for an extended version of AgentSpeak. Jason implements the operational semantics of the AgentSpeak language, and provides a platform for the development of multi-agent systems, with many user-customisable features [38]. *AgentSpeak* is an agent-oriented programming language based on logic programming, and inspired by the work on the belief-desire-intention (BDI) architecture [31]. The BDI architecture is regarded as the predominant approach to the implementation of intelligent or rational agents [32]. The detailed Jason programming approach is elaborated in [33]. AgentSpeak is based on description logic rather than predicate logic (imperative programming) [34]. Employing description logic avoids the problems in imperative modelling, as declarative models specify what should be done without specifying how it should be done [7]. In view of this the knowledge layer of the simulation is implemented in Jason. The use of description logic has another advantage that it is at the core of widely known ontology languages (including OWL) which are used to share knowledge with agents. In this way more sophisticated reasoning is facilitated [34]. The last advantage of using Jason is that it can be uses in the JADE environment. As an infrastructure it can make use of facilitating services provided by JADE, in particular the DF service.

We found only one implementation of an agent-oriented programming language employing transparent use of ontologies and underlying ontological reasoning within a declarative setting. This implementation named JASDL (Jason AgentSpeak–DescriptionLogic), uses Jason customisation techniques and the OWL-API [35] in order to provide all the features of agent programming combined with ontological reasoning [36]. JASDL is available as open source software.

4 Illustrating the Approach in Practice

The agent-based simulation was used to evaluate if changes in legislation could be accomplished by separating knowledge rules, process rules and information resources. Within the AGILE project the Dutch Immigration and Naturalisation Service (Immigratie en Naturalisatiedienst, IND) is the partner that provides the empirical setting for our experiments. The IND is the organization that handles the admission of foreigners in the Netherlands. The IND is responsible for the execution of a complex set of regulations in this domain coming from different sources of law, including international law, national law, case decisions and policy directives. Moreover, the IND makes a large number of decisions, i.e. some 300,000 a year in the areas of asylum, standard objections to decisions, and naturalisation. Under pressure from frequently changing law and regulations, the IND is one of the governmental organizations that desire a strong assurance of compliance in their IT system. As a research partner in AGILE project, the IND provides a practical context for our research.

A use case dealing with a highly skilled migrant that applies for a residence permit was selected to prove our concepts. One of the reasons for choosing this use case is that it is often subject to changes (initiated at the politician levels) and has a large number of applicants. The highly skilled migrant admission legislation is introduced to enable qualified foreigners to work in the Netherlands. The policy with respect to highly skilled migrants has been changed frequently in the recent years. In 2007, the annual income limitation on an applicant is at least 46,541 or 34,130 EUR if the applicant is under 30. In 2008 there was a change in policy in order to retain foreign intelligent graduates and encourage them to work in the Netherlands, The income limitation for the foreign graduate that obtained a Bachelor or Master Degree at an accredited Dutch educational institution within one year before becoming employed was changed to 25,000 EUR annually.

In 2009 there was another change in legislation. Employees must have a gross annual income of at least 49,087 EUR or 35,997 EUR if they are under the age of 30. There are two different situations to which the reduced wage criteria applies (in 2009 25,800 EUR gross a year). The first situation is aimed at graduates that obtained a Bachelor or Master Degree at an accredited Dutch educational institution, similar to the 2008 situation. The second situation concerns Master and PhD students who graduated in the Netherlands or at a university listed in the top 150 of two internationally recognized rankings.

There are basically three types of changes that needed to implement to ensure compliance to the legislations.

1. *General income requirements*: The general income requirements need to be changed. This occurs frequently as the wage criterion for highly skilled migrants is indexed annually.
2. *Additional check for educational institutions*: A check for Bachelor and Master degrees at accredited Dutch educational institutions has to be added to the admission process.
3. *Additional check for ranked institutions*: A check for Master and Ph.D. degrees at certain ranked educational institutions has to be added to the admission process.

Flexibility is needed to comply with these changes in legislation. The first type of changes requires only a change in income. In the second and third kind of changes are more difficult as the IND needs to check the applicant's educational background which has a lot more impact on previous business processes and information systems. In the new situation applicants are required to send in a proof of their educational background, and the IND needs to introduce a control to verify whether this proof is valid or not. The admission process has to be changed and new tasks and new objects have to be added. All three changes have an impact on the number of admissions requests from foreigners and the processes should be agile enough to react to changes. The simulation of the three changes and their impact are discussed hereafter, and a part of the Jason code is presented to illustrate the implementation of agents.

4.1 Change in Income Requirement

The admission process at IND is started when a client submits a proper application for a residence permit together with all the necessary information (see C1). One of the available processor agents will receive this application and responds to the client that he is responsible for dealing with the application and will inform the client agent about the final decision after the application has passed all the necessary steps. The behaviour of the Client agent could be as follows:

C1: Client agent

```
//initial goal

!sub-
mit_application(application(applicant,application_type)
).

/*Plans*/

+!submit_application(application(applicant,
application_type))
 : ture & available_processor(P) & P>0
 <- .send (proces-
sor,achieve,application(applicant,type)).
```

The next step for the Processor agent is to figure out what kind of application has been submitted. This process is similar to that of an employee who would receive a closed envelope with an application in it. The system does not know what kind of application has come in. The Processor agent will then consult the Process Rule Manager to receive information with regard to the type of application and the next step, based on the type of application (see C2 and C3). Again this is done by consulting the modelled information concerning the business process. In our example we are dealing with a client that wants to apply for a residence permit as a highly skilled migrant. If the application is processed, the next steps are rather easy for the situation in 2007. The only two things that need to be checked are the migrant's age and the income the migrant will receive at his future employer. This information will be checked by the Knowledge Rule Manager agent against the knowledge model (see C4). If the income meets the specified income limit (according to the specified age), the decision of the system will be that the residence permit will be granted. This information is communicated back to the Processor agent that in turn will communicate it to the Client.

C2: Processor agent

```
/*Plans*/

+receive_application(application) : true
 <- .creat_case(case);
 !get_next_step(case).

+!get_next_step(case):true
 <- .send(process_rule_mananger,acheive,case)

+receive_indication(next_step(receiver,case)) : true
 <- .send(receiver,acheive,case).
```

C3: Process Rule Manager agent

```
/*Plans*/

+receive_case(source(Sender),case) : true
 <- .creat_step(step(receiver,task));
 .send(Sender,tell,step(receiver,task)).
```

C4: Knowledge Rule Manager agent

```
/*Rules*/

granted(A) :-
 .age(A,Age)&
 .income(A,Income)&
 (Age >= 30 & Income >= 46541) |
 (Age < 30 & Income >= 34130).
```

This example demonstrates that by using the Knowledge Rule Manager agent, changes on numbers, such as the income requirements, will not affect the process and compliance can be ensured by adapting the model behind the Knowledge Rule Manager. Flexibility concerning changes in income is created in this way. The dynamic creation of business processes by looking at available processor agents ensures agility. In case of slow response times more processor agents need to be created.

4.2 Additional Check for Dutch Educational Institutions

The first part on how this situation will be processed is quite similar to the previous case. Therefore we will start describing from the moment the processing differs from the previous case. As soon as the Processor agent reaches the Process Rule Manager agent, the change of process steps is noticed. An extra step is needed to check for a Bachelor or Master degrees at an accredited Dutch educational institutions. The question is who can provide the service of checking such a degree. The Processor agent will consult the Resource Rule Manager agent to find out whether there is an existing service that can be used for this purpose (see C5). The Resource Rule Manager agent is aware of the source of information, such as which agent provides the wanted service, or whether the information is provided by human input or by a software application (e.g. a web service). The model that is behind the Resource Rule Manager agent and can be accessed through the API should be updated to provide information with respect to this extra process step. For example if the check on Dutch educational background can only be done by human staff, this should be added to the Resource model. If there is a connection to educational organizations possible this should be added here, to make sure the system knows what or whom to address to perform this check. Often it is possible to perform this check automatically, by making contact with a database that contains information on bachelor and master degrees obtained at Dutch higher educational institutions. In our architecture an External Agent will represent the service of checking. After the check has been done, the Processor agent will get the result back from the External Agent. Then the Processor will consult the Process Rule Manager again to find out about the next step. If all necessary information is there, the Process Rule Manager will lead the Processor to the Knowledge Rule Manager, to make the final decision. The final decision is communicated back to the Client agent by the Processor agent.

C5: Resource Manager agent

```
check_degree(External_Agent_ID).//initial belief

/*Plans*/

+receive_case(source(Sender),task) : true
 <-
.creat_resource(resource(provider,resource_notation));
.send(Sender,tell,resource(provider,resource_notation).
```

When comparing the simulation of this case with the previous one, the advantages of using the Process Rule Manager become clear. An extra step can be added to the process model to ensure compliance to the changed legislation, without having to make huge changes to the whole process. The only work that has to be done is adapting the model behind it and adding data of new information sources into the model used by the Resource Rule Manager. The modifiability dimension is improved as new processes can be added. Furthermore the partnership dimension of flexibility is improved as the services offered by external agents can easily be added to the process.

4.3 Additional Check for Certain Ranked Educational Institutions

This situation is highly similar to the previous ones, therefore we will only describe this case from the moment it is different from the previous case. The difference is the extra step that is needed in order to check for a degree at a ranked educational institution. Again the question is who can provide the service of checking for such a degree. The Processor agent will therefore consult the Resource Rule Manager to find out whether there is an existing service that can be used for this purpose. Again the model that is behind the Resource Rule Manager should be updated to provide information with respect to this new process step. Now we assume that this step cannot be performed automatically, instead we need human staff for it. In our architecture a hu man computer interface can be provided to facilitate a manual check of the ranked institutions. After the check has been done, the Processor agent can continue executing its remainder tasks. Thereafter the Processor agent will consult the Process Rule Manager agent again to find out about the next step. If all necessary information is available, the Process Rule Manager agent will lead the Processor agent to the Knowledge Rule Manager agent, to make the final decision. The final decision is communicated back to the Client by the Processor agent.

If we compare the simulation of this example with the one described in the first subsection, it clearly points out the advantage of using the Resource Rule Manager. A step in the process has changed; however, we only need to adjust the required resource information instead of having to change the process itself. In this way, the compliance to the changed legislation can be ensured effectively and efficiently. Furthermore, a shift from an automatic processing step to a manual processing step can be facilitated. Involving a human intervention in the process is just as easy as involving other automatic steps. From this case, we can see that manual assistance is required to take care of a certain part of the process. It is highly likely that this will require much more effort and time compared to an automatic check; therefore organizations are already prepared to the fact that a higher overall time is needed to complete the changed process as a whole or more staffs is needed to deal with a case from start to finish. The same type of flexibility is created as in the previous example.

5 Conclusions and Future Research

Flexible and agile information systems and business processes are the key for any organization to deal with the changes in its dynamic environment. In this paper a mechanism based on the separation of knowledge rules, process rules and information resources is proposed to enhance the flexibility and agility of business processes. The separation of these elements proved to be helpful to be able to react to changes in legislation.

Agent-based simulation is used to test if these concepts enhance the flexibility and agility. Agent-based simulation helps us to understand the impact of measures for the realization of flexibility and agility. A real life case study was used to test the approach. Three situations are modelled and the results showed that by using the Knowledge Rule Manager, changes on numbers, such as the income requirements, will not affect the process and compliance can be ensured by adapting the model behind the Knowledge Rule Manager. Furthermore an extra process step can be added by updating the process model and there is no need for having to make changes to the whole process. The modifiability dimension of flexibility is improved as new processes can be added. Finally, the partnership dimension of flexibility is improved as the services offered by external agents can easily be added to or removed from the process. Agility is improved primarily by facilitating the management of knowledge rules, process rules and information resources. The simulation results of the scenarios show that the separation of knowledge rules, process rules and information resources is a usable mechanism to enhance the flexibility and agility of business processes.

Our approach employs agent-based simulation to assess the impact of changes in order to be able to predict how an organization's business processes and applications will respond. In that way certain measurements to ensure flexibility and agility can be made and tested.

In further research OWL axioms and rules can be added to include real-life rules. Existing work [37] that is aimed at classification of Dutch legislation, e.g. norm sentences, can be used as a first step towards automated modelling of the law. It would be interesting to see whether we can apply that approach in a similar way to automatically extract information and in this way create models of services out of law and regulations.

We have introduced our concept on how to combine both the area of OWL and agent-based simulation and what the benefits of this approach are. However, at this moment there is no connection yet between the knowledge representation of the law, legal rules, business processes and services and the part that covers the agent-based simulation. So far we have only experimented with the JADE and Jason agents and relevantly simple scenarios with limited number of interactions between agents. A proper connection between JADE and Jason, as well as, Jason and an OWL editing and reasoning environment, for example Protégé, is one of the main research directions that needs to be researched. We expect that this extension will enable more complex agent behaviours and interactions.

Acknowledgement. This work is supported by AGILE project (Advanced Governance of Information services through Legal Engineering, detail can be found on the web page http://www.jacquard.nl/?m=426).

References

1. Gong, Y., Janssen, M., Overbeek, S.J., Zuurmond, A.: Enabling flexible processes by ECA orchestration architecture. In: Janowski, T., Davies, D. (eds.) The 3rd International Conference on Theory and Practice of Electronic Governance (ICEGOV 2009), pp. 19–26. ACM Press, New York (2009)
2. Georgakopoulos, D., Hornick, M., Sheth, A.: An overview of workflow management: From process modeling to workflow automation infrastructure. Distributed and Parallel Databases 3, 119–153 (1995)
3. van der Aalst, W.M.P., ter Hofstede, A.H.M., Weske, M.: Business Process Management: A Survey. In: van der Aalst, W.M.P., ter Hofstede, A.H.M., Weske, M. (eds.) BPM 2003. LNCS, vol. 2678, pp. 1–12. Springer, Heidelberg (2003)
4. Li, L., Zhao, X.: Enhancing competitive edge through knowledge management in implementing ERP systems. Systems Research and Behavioral Science 23, 129–140 (2006)

5. McGaughey, R.E.: Internet thechnology: contributing to agility in the twenty-first ecntury. International Journal of Agile Management Systems 1, 7–13 (1999)
6. Wadhwa, S., Rao, K.S.: Flexibility and agility for enterprise synchronization: knowledge and Innovation Management Towards Flexagility. Studies in Information and Control 12, 111–128 (2003)
7. Pesic, M., van der Aalst, W.M.P.: A Declarative Approach for Flexible Business Processes Management. In: Eder, J., Dustdar, S. (eds.) BPM Workshops 2006. LNCS, vol. 4103, pp. 169–180. Springer, Heidelberg (2006)
8. Heinl, P., Horn, S., Jablonski, S., Neeb, J., Stein, K., Teschke, M.: A comprehensive approach to flexibility in workflow management systems. In: The International Joint Conference on Work Activities Coordination and Collaboration, pp. 79–88. ACM, New York (1999)
9. Janssen, M., Verbraeck, A.: An Agent-based Simulation Testbed for Evaluating Internet-based Matching Mechanisms. Simulation Modelling Practice and Theory 13,, 371–388 (2005)
10. Russell, S.J., Norvig, P.: Artificial Intelligence: A Modern Approach. Prentice Hall, Upper Saddle River (2003)
11. Wu, J.: An Agent-based Simulation Study for Exploring Organizational Adaptation. Simulation 85, 397–413 (2009)
12. Alter, M.J.: Science of flexibility. Human Kinetics (2004)
13. Carlsson, B.: Flexibility and the theory of the firm. International Journal of Industrial Organization 7, 179–203 (1989)
14. Gosain, S., Malhotra, A., Sawy, O.A.E.: Coordinating for Flexibility in e-Business Supply Chains. Journal of Management Information Systems 21, 7–45 (2004)
15. Tan, C., Sia, S.K.: Managing Flexibility in Outsourcing. Journal of the Association for Information Systems 7, 179–206 (2006)
16. Bernardes, E.S., Hanna, M.D.: A theoretical review of flexibility, agility and responsiveness in the operations management literature. International Journal of Operations & Production Management 29, 30–53 (2009)
17. Kumar, A., Motwani, J.: A methodology for assessing time-based competitive advantage of manufacturing firms. International Journal of Operations & Production Management 15, 36–53 (1995)
18. Vokurka, R.J., Fliedner, G.: The journey toward agility. Industrial Management & Data Systems 98, 165–171 (1998)
19. Gunasekaran, A.: Agile manufacturing: a framework for research and development. International Journal of Production Economics 62, 87–105 (1999)
20. Meredith, S., Francis, D.: Journey towards agility: the agile wheel explored. The TQM Magazine 12, 137–143 (2000)
21. Bieberstein, N., Laird, R.G., Jones, K., Mitra, T.: Executing SOA: A Practical Guide for the Service-Oriented Architect. Addison-Wesley, Reading (2008)
22. Medina-Mora, R., Winograd, T., Flores, R., Flores, F.: The action workflow approach to workflow management technology. In: ACM Conference on Computer-Supported Cooperative Work, pp. 281–288. ACM, New York (1992)
23. Faget, J., Marin, M., Patrick, M., Owens, V.J., Tarin, L.-O.: Business Processes and Business Rules: Business Agility Becomes Real. Future Strategies Inc. (2003)
24. Lienhard, H., Künzi, U.-M.: Workflow and Business Rules: a Common Approach. In: Workflow Handbook 2005, pp. 129–140. Future Strategies Inc. (2005)
25. Müller, R., Greiner, U., Rahm, E.: Agent Work: a workflow system supporting rule-based workflow adaptation. Data & Knowledge Engineering 51, 223–256 (2004)

26. Law, A.M., Kelton, W.D.: Simulation Modeling and Analysis. Mc-Graw Hill, New York (2000)
27. Wetzel, I., Klischewski, R.: Serviceflow beyond workflow? IT support for managing inter-organizational service processes. Information Systems 29, 127–145 (2004)
28. Likert, R.: The human organization: its management and values. McGraw-Hill, New York (1967)
29. Zhao, Z., Belloum, A., De Laat, C., Adriaans, P., Hertzberger, B.: Using Jade agent framework to prototype an e-Science workflow bus. In: The Seventh IEEE International Symposium on Cluster Computing and the Grid., pp. 655–660. IEEE Computer Society, Washington (2007)
30. Yoo, M.-J., Glardon, R.: Combining JADE and Repast for the Complex Simulation of Enterprise Value-Adding Networks Agent-Oriented Software Engineering IX, pp. 243–256. Springer, Heidelberg (2009)
31. Bordini, R.H., Hübner, J.F.: BDI Agent Programming in AgentSpeak Using Jason. In: Toni, F., Torroni, P. (eds.) CLIMA 2005. LNCS (LNAI), vol. 3900, pp. 143–164. Springer, Heidelberg (2006)
32. Wooldridge, M.: Reasoning about Rational Agents. The MIT Press, Cambridge (2000)
33. Bordini, R.H., Hübner, J.F., Wooldridge, M.: Programming Multi-Agent Systems in AgentSpeak using Jason. Wiley, Blackwell (2007)
34. Moreira, Á.F., Vieira, R., Bordini, R.H., Hübner, J.F.: Agent-Oriented Programming with Underlying Ontological Reasoning. In: Baldoni, M., Endriss, U., Omicini, A., Torroni, P. (eds.) DALT 2005. LNCS (LNAI), vol. 3904, pp. 155–170. Springer, Heidelberg (2006)
35. Horridge, M., Bechhofer, S.: The OWL API: A Java API for Working with OWL 2 Ontologies. In: 6th OWL Experienced and Directions Workshop, CEUR Workshop Proceedings, Chantilly, Virginia (2009)
36. Klapiscak, T., Bordini, R.H.: JASDL: A Practical Programming Approach Combining Agent and Semantic Web Technologies. In: Baldoni, M., Son, T.C., van Riemsdijk, M.B., Winikoff, M. (eds.) DALT 2008. LNCS (LNAI), vol. 5397, pp. 91–110. Springer, Heidelberg (2009)
37. de Maat, E., Winkels, R.: A next step towards automated modelling of sources of law. In: International Conference on Artificial Intelligence and Law, pp. 31–39. ACM, New York (2009)
38. Hübner, J.F., Bordini, R.H.: Jason Home page (2009)

Using Fact-Based Modeling for Business Simulation

Peter Bollen

School of Business and Economics
Maastricht University, Tongersestraat 53,
6200MD, Maastricht, The Netherlands
p.bollen@maastrichtuniversity.nl

Abstract. Fact-Based Modeling (FBM) is a conceptual modeling language that can be used for requirements determination for business application systems. In this article we will show how the Fact-based Modeling methodology for requirements determination can be extended to serve as a blueprint for business simulation by providing an initial model for creating a business simulation. We will do this by defining the content of the communication documents for run-time management and we will subsequently show how this meta-UoD can be incorporated into an application UoD. This allows us to capitalize on conceptual models in a business that have been created for requirements determination by extending them with the conceptual model of runtime management. Subsequently, we will incorporate the simulation requirements into the latter UoD and we will give guidelines on how conceptual models for the 'real-life' runtime application can serve as a starting point for the conceptual sub-models for the simulation UoD.

Keywords: simulation and modeling IS, design methodologies, information analysis techniques.

1 Introduction

There's a growing number of articles in the simulation field that discuss the interdependence between a conceptual model (or operational model) of a subject area and a simulation model [7, 10, 13, 27, 28]. In parallel other authors indicate the interdependence between the simulation data sources and the operational business data sources that are contained in the corporate business systems [20, 21]. These authors commonly identify the need for an integrated approach towards (the design of) business simulations and the (design of) operational information systems. Another stream of research is concerned with the definition of a methodology for business (process) simulation [9, 12, 25] that can be used for assessing potential 'to-be' business process designs [8] and in which structural validation of a simulation with the actors in the system is advocated [2]. Nidumolo et al. [16] discuss the application of information systems architecture-based (ISA) approaches for object-oriented modeling and simulation (OOMS) in which they conclude that the strong roots of these approaches in static IS modeling amongst other factors limits them to be used for business process simulation. In this paper we will challenge Nidumolu et al.'s position and take the application of ISA approaches for business simulation, one step further by providing

J. Barjis (Ed.): EOMAS 2010, LNBIP 63, pp. 59–78, 2010.

an integrated framework for the three levels (information base, schema and meta-schema) and three perspectives (information, process and behavioural) for information systems [18]. We will apply this framework on an integrated Universe of Discourse, by defining the communication documents for runtime management and by incorporating the simulation requirements into the application requirements documents. A requirements determination approach in which a strong modeling methodology is contained that has been developed within the architecture of this framework is fact-based conceptual modeling of which CogNIAM [15, 19] and ORM [11] are popular dialects.

A fact-based information grammar contains a domain ontology and all fact types that govern the communication in a specific subject area. In addition the business rules that govern the communication, i.e. stating which fact instances are now allowed to exist in combination can be expressed as population constraints.

2 The Fact-Based Constructs for the Process Perspective

In this section we will define the modeling constructs for the process-oriented perspective that can be applied within business subject areas. The process perspective in an enterprise subject area is concerned with 'how' fact instances can be composed from other fact instances. In businesses we can consider facts as either an *outcome* of an enterprise activity or an *ingredient* for an enterprise activity. Enterprise activities are executed under the responsibility of a *user* from a *user group*. We will call a *user* that creates facts, an *active user*. The border concept in the process perspective will show what *user groups* can be held responsible for the creation of fact instances in the UoD.

Definition 1. A *conceptual process instance* is the abstraction of an organizational activity that is responsible for the creation of (a) *fact instance(s)* by an *active user*.

2.1 Derivation Process Types

The fact type(s) of the fact instances created in (an) instance(s) of a conceptual process type will be referred to as the *resulting fact type(s)* for the conceptual process type. An (the) *ingredient fact type(s)* of a conceptual process type specifies what the fact type(s) is (are) for the fact instances that serve as an input for the creation of a fact in a process instance of a given conceptual process type. In case the 'underlying mechanism' is a procedure or a derivation rule (see figure 6) on a descriptive document that specifies for *all* instances of the conceptual process type how the resulting fact instance(s) (contained on a declarative document) can be derived from the ingredient fact instances we will call such a conceptual process type a *derivation process type*.

Definition 2. A *derivation process type* is a conceptual process type whose process instances create fact instances by applying the same derivation rule on instances of the same ingredient fact type(s).

2.2 Determination Process Types

Some facts will be created without a known (or existing) derivation rule. For example the creation of the first name of a new-born. However, in many cases the creation of such a fact is subject to constraints. In the example of the name assignment for a newborn, the following constraint exists: a baby of the female gender must be assigned a girl's name and a baby of the male gender must be assigned a boy's name (from a predefined list of names).

Definition 3. A *mixed determination process type* is a conceptual process type in which the active user uses instances of the same ingredient fact types (that are contained in the application's information grammar) in all process instances.

The conceptual process that creates the names of a newborn baby: *We have decided to call you John. We have decided to call you Alice.* These examples do not involve any derivation rule or (formal) procedure, but it is assumed that ingredient fact instances exist, for example: *John is the name for a boy, Alice is the name for a girl, The child that should be named is a girl* must be known, before a name can be created for a specific child. The way in which a name is assigned in a specific instance, however, can not be determined in advance. Some people might select the name of their own father or mother for their child. Others might choose the name of their favourite rock star. On a 'process type' level, however, we can never know what selection criterion (or derivation rule), will be applied in a specific process instance. The same parent might use different criteria for every newborn. In addition to *derivation* and *mixed-determination* process types we can distinguish conceptual process types which have no known and fixed set of ingredient fact type(s) and derivation rules: *strict-determination process types*. This type of proces is used in managerial decision making, for which, in some cases, *decision support systems* are employed: "The user may only need 40-100 data variables, but they must be the right ones; and what is right may change from day to day and week to week." [26, p.21].

Definition 4. A *strict-determination process type* is a conceptual process type in which the active user does not use a known derivation rule all the time and the active user does not use instances of the same ingredient fact types (that are contained in the application's data model) in all process instances.

Definition 5. A *conceptual process type argument* specifies the types of values that must be supplied for instantiating a conceptual process.

When we consider the derivation process type *determine customer credibility* in figure 2, it will only create (a) fact instance(s) of fact type FT3 when at least one fact instance of fact type FT2 exists in the *application information base (AIB)* in which the value for the role 'customer' is equal to the value for the process argument 'arg1'. If we inspect the derivation rule for this conceptual process type and the instantiation values for the *process type argument* it should be clear whether the execution of the process will lead to a result **before** the derivation rule is actually executed or fact instance(s) are determined by a active user. The *pre-condition* for a conceptual process type serves as checking mechanism for the instantiation of a process type (figure 1). If the *pre-condition* is violated by the actual content of the *AIB*, the process will **not** be executed and (a) resulting fact instance(s) will **not** be created.

Definition 6. A *precondition* in a conceptual process type checks whether the required *input fact instances* for the *derivation process* or *the mixed determination process* exists in the *enterprise data base.*

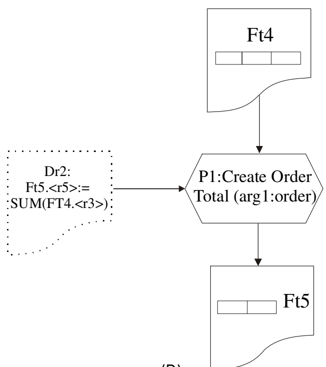
The post-condition specifies what the fact argument is for the facts that will be created in the conceptual process (see figure 1). Furthermore, it is specified *how* the fact values that will be created in the conceptual process will be obtained. In case of a *derivation process* a reference is given to a derivation rule. In case of a *mixed-* or *strict-* determination process, it is stated that (a) fact(s) has (have) to be created (by an active user). This post-condition, furthermore, specifies how the resulting fact type(s) of the process type, must be instantiated as a function of the instantiation values for the conceptual process type argument.

Definition 7. A *post-condition* of a conceptual process type specifies (parts of) the fact argument for the instances of the resulting fact type for the conceptual process. A *post-condition* in a conceptual process indicates that (a) fact value(s) ha(s)ve to be determined. A *post-condition* in a derivation process type, furthermore, specifies what derivation rule is used for the creation of the resulting fact instance(s).

We will now simplify the specification of a conceptual process type by dividing such a specification in (at most) 3 parts. In the case of a *derivation process type* we will specify a *precondition*, a *postcondition* and a *derivation rule*. In case of a *mixed-determination process type* we will specify the *precondition* and *postcondition* and, finally, in case of a *strict-determination process type* we will only specify the *post-condition*.[1]

P1: Create order total (arg1:order)
precondition instance of Ft4 exists
where FT4.<R2>= 'arg1'
postcondition create instance of Ft5
where FT5.<r4>= 'arg1'
and FT5.<r5>:= Dr2
derivation rule Dr2: SUM(FT4.<r3>
[where FT4.<r2>= 'arg1'])

(A)



(B)

Fig. 1. (a) Textual process type representation and (b) Accompanying graphical representation

For some fact types in an *application information grammar* (AIG), the conceptual processes that create fact instances are performed under the responsibility of users outside the application's sphere of influence. We will by default define such fact creating process type as an *enter* process type. In figure 1 we have given the textual *application process description* (APD) together with a graphical representation. An

[1] We are only interested in conceptual process configurations in which an instance of a given fact type is created. Therefore, in our view only process configurations having a post-condition are relevant. Furthermore, we can only define a specific derivation rule when a post-condition exists.

elaborate description of the methodology for the process-oriented perspective can be found in [4].

3 The Fact-Based Constructs for Modeling the Event Perspective

Although the execution of conceptual processes is constrained by the *pre-conditions*, *post-conditions* and *derivation rules* from the respective conceptual process types there still remain degrees of freedom with respect to *when* and in *what sequence* these conceptual processes can be executed. In some business situations the compliance to the *AIG* and the *APD* is sufficient for enforcing the business rules in the application area. In other application areas additional modeling constructs are needed that can specify ***when*** the instances of conceptual process types from the *APD* will be executed.

Example . Consider the following example of a workflow management application (see figure 2).

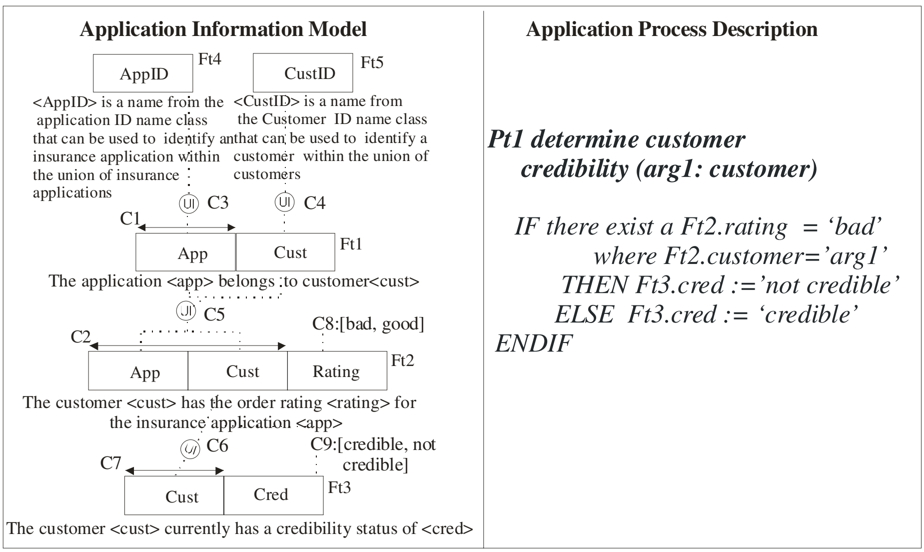


Fig. 2. AIG and APD for example

Whenever an *insurance application* is created an instance of the process type *credibility checking* must be executed:

ON *insurance application is created*
IF *credibility checking capacity is available*
THEN *perform credibility checking*

The description of the activity that is specified in the ON clause in the above example can be considered an event or occurrence of something that has happened in the application subject area.

3.1 Event and Event Type

In the information systems literature numerous definitions of the *event* concept can be found*: "An event is an occurrence or happening of something in the environment under consideration."* [6, p.182]. *"An event is a noteworthy change of state; all the changes of state of objects are not events."* [22, p.34]. *" When the environment does something to which the system must respond, an event is said to occur."* [29, p.198]. *"An event is a happening that changes the state of a model (or system)."* [24,p.15]. We will now give the following definition of *event*:

Definition 8. An event is an occurrence or happening in the application subject area that can lead to the execution of one or more conceptual processes within the application subject area's sphere of influence.

In the insurance application subject area we have to specify whose insurance application request is created. We therefore have to *qualify* the example event from the insurance applocation example: *Insurance application is created* into the following event: *Insurance application with application number 45678 is created.* If we observe this application subject area over a certain period of time we can encounter also the following event instances *Insurance application with application number 45679 is created, Insurance application with application number 45680 is created.* We can conclude that the former verbalization of events can be further grouped and qualified into the *event type*: *Insurance application created (arg1: application number)*

Definition 9. An event type is a class of events that have the same intentions according to a user group *G* for an universe of discourse *UoD* and a sphere of influence *SoI*.

In order to make a distinction into fact types and event types we will model the *'role(s)'* in an event type as *event argument(s).* The intention that plays such a role in an event type in principle is defined in the same way as the intention concept in the information perspective. A specific intention that is defined in an event type, however, does not necessarily have to be defined in the AIG of that subject area. We will now formalize the intentions(s) of the event type by structuring these(is) intentions(s) into the *event type argument set*. We will derive the set of arguments for the event type by *classifying* and *qualifying* a significant set of verbalizations of event instances, for example:

Insurance application 257892 is created.
Insurance application 28923 is created.

This will result in the following event type and its argument set:

Insurance application created (arg1: application)

Definition 10. An *event type argument set* of a given event type specifies all intentions, instances of which should be supplied at the occurence of an event instance of the event type.

Consider the following event type:ET1: *Customer at table wants to pay (arg1:customer)*

An *instance* of this *event* type is: *Customer at table wants to pay (arg1:'Piet Janssen').*

The conceptual process that should be instantiated as a result of this event is an instance of the following conceptual process type: *P1: determine order total (arg1:order)*. The instances of the intentions in the argument set for the event type can be used for instantiating the process type (in an *impulse*). The modeling construct of *event* refers to an action that can occur, for example: *customer places an order* or it can refer to a more 'static' action, for example the start of a new day when *the clock strikes 12:00 P.M*. We can conclude that events can have different appearances and therefore we will use the *AIG* and the *APD* in combination as a starting point for 'detecting' events that are relevant for the application subject area which means that all events that do ***not*** potentially trigger a conceptual process will be left out.

3.2 Event Condition and Event Condition Type

An *event* can start the execution of a process (in some cases) under (a) condition(s) on the information base. In the population constraints from the AIG we have modeled the business rules that are always applicable (or invariant) in terms of the AIB. For example the business rule that states that *each order has at least one orderline*. In the pre-conditions for the conceptual processes in the application process decription, the business rules are modeled that specify what ingredient fact instances should be available in order to 'compose' or 'derive' the resulting fact instance(s) in the conceptual process. In the event perspective we will model the business rules that contain the knowledge under what condition (on the AIB) the occurence of an *event* of an *event type* will trigger a *process instance* of a specific *process type*.

An *event condition* is a *constraint* for the execution of a conceptual process that is 'triggered' by a specific event. In addition to the conditions that are given in the *pre-condition* of the conceptual process or a condition that is enforced by the *population constraints* in the AIG, the condition that is specified in the event perspective is defined in terms of the *AIB* at that moment in relative time in which the *event condition* is checked.

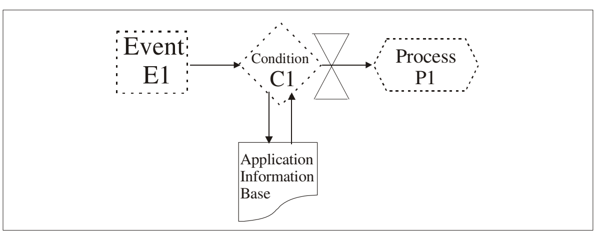


Fig. 3. Event, condition and process

Definition 11. An *event condition* is a proposition on the information base.

Example: c_1: $\exists_{f \in EXTENSION(FT1)}[f.<r2>= 'Piet']$

In most business application areas the occurence of an event instance at t_1 should lead to the execution of a conceptual process (see figure 3). The occurence of the same event instance at another moment in relative time should not necessarily lead to the execution of a conceptual process because the *AIB* can be in a different state.

Example:
Event type *order request(arg1: customer, arg2:product)*
Event instance *order request (arg1:"Piet Janssen" , arg2: "Bicycle").*

Under the condition that the customer has a satisfactory credit-rating this event should lead to the instantiation of the following conceptual process type: *prepare order confirmation (arg3: customer, arg4: product).* The instantiation will lead to the following process: *prepare order confirmation (arg3: "Piet Janssen", arg4: "Bicycle").* A different event occurence of the same type is *order request (arg1: "Hans Koek", arg2: "Scooter").* Given the fact that this customer in this case does not satisfy the credit rating, the event occurence should **not** lead to the instantiation of (a) conceptual process(es). In most business application areas it is possible to specify these conditions on a type level: *" If a customer wants to order a product he/she should have a sufficient credit rating."* A condition (instance) of the forementioned condition type is: *" If Piet Janssen wants to order a product he/she should have a sufficient credit rating."* It can be seen that the instantiation values for the event type can be used in this case for the instantiation of a condition on the information base.

Definition 12. An *event condition type* is a set of propositions defined on the information base. An instance of a condition type: a condition can be derived whenever a(n) event instance is specified.

Example:
ET1: Customer orders an order (arg: customer)
CT: $\exists x \in EXTENSION(R_2)\ [\ x=ET1.arg]$

3.3 Impulse and Impulse Type

We will call the effect of an event occurence into the execution of one conceptual process (eventually under an event condition on the information base) an *impulse* (instance).

Definition 13. An *impulse instance* is the potential triggering of the execution of a *conceptual process instance(s)* from the *APD* dependent upon the condition in the *AIB*.

Example:(order 56 is delivered, (stocklevel <125), P_2*)*

Events that do not have the potential to 'trigger' conceptual processes from the APD are not relevant for the description of the behavioural perspective in a given application subject area. We can now classify all impulses that have the same *event type*, the same *process type* and the same *event condition type* into a set of impulse instances that belong to the same *impulse type*.

Definition 14. The intention of an *impulse instance* is an *impulse type*.

We will now introduce a construct in the event perspective that enables us to derive the instantiation value for the process argument in an impulse whenever the values of the *event type argument set* are known. This is the construct of an *impulse mapper.* The impulse mapper is a mechanism that encodes the business rules in the subject area that specify *how* a conceptual process is instantiated when an event occurs and the condition on the information base (specified in the impulse) is satisfied.

Definition 15. An impulse mapper is a construct that transforms values of event type arguments and fact instances from the AIB into instantiation values for the *process type argument set(s)* for the process type(s) that will be potentially instantiated in the impulse.

Event type	*Et1: insurance application created (arg1: application).*
Process type	*Pt1: determine customer credibility (arg1:customer).*
Condition type	*Ct1: ET1.arg1 ∈ EXTENSION(person3)*
Impulse type	*IT1:=<Et1, Ct1, Pt1>*
Impulse mapper	*Pt1.arg1:=Ft1.cust (where Ft1.app='Et1.arg1')*

We still need a modeling concept that allows us to express the 'triggering' of conceptual processes whose types are ***not*** contained in any of the impulses of the *application event description* (AED)so far. This concept will be the *trigger-process* event type. An occurence of such an event type will immediately and unconditionally result in the instantiation of the conceptual process type that is specified. In the *trigger-process* event type the argument set of the event type is equal to the argument set of the process type. Including such a *non-conditional* and *unqualified* impulse in an event description implies that such a conceptual process can be instantiated at any time and it therefore implies that no constraints on the behavioural perspective exist for the instantiation of such a process type. In addition we need to time variables that can encode the potential time delays between on the one hand the occurrence of an event and the check on the event-condition (in the impulse) and on the other hand the check on the condition and the execution (if any) of the conceptual process (see [3, p.207-208]).

3.4 The Fact-Based Event Modeling Methodology

In addition to the AIB and the AIG in the information perspective and the APD in the process perspective we will define the AED as the document that constrains the knowledge behaviour for an application subject area in the event or behavioural perspective. In [3, p.162-163] we have specified the procedure that an analyst must apply in order to yield the AED. A more elaborate treatment of the modeling constructs and a description of the modeling methodology in the behaviour-oriented perspective in the event perspective can be found in [3: p. 161-165; 5: p. 199-209].

4 Extending the Subject Area with Runtime Management

The conceptual documents we have defined so far (see figure 4) *intentionally* constrain the communication and 'knowledge' behaviour within an organizational subject area. In this section we will introduce an additional *universe of discourse* and *sphere of influence* for the application areas in our architecture. This specific UoD and SoI will be needed for enforcing the *extensional* compliance to the *fact types* and the population *constraints* in the *AIG*, the *process types* in the *APD* and the *impulse types* in the *AED*. We will call this additional subject area: *runtime management*. The sphere of influence of runtime management contains three user groups: the *event manager,* the *process manager* and the *information base manager*. We will define the UoD of runtime management as the union of 'real-life' examples that are used by these user groups.

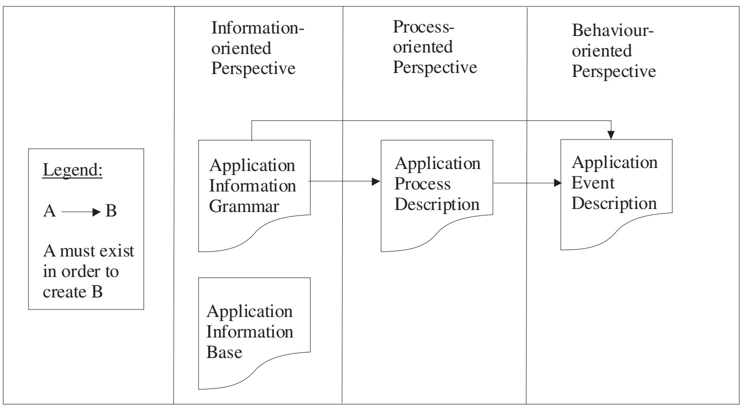


Fig. 4. The four conceptual documents for an application subject area

4.1 The Event Manager User Group

We will now introduce a conceptual agent that will evaluate the occurence of events during runtime. This conceptual agent will detect the *occurence* of an event, determine its *event type*, evaluate the *event condition* in the *impulse* relative to the information base at the moment of evaluation and invoke the relevant *conceptual process types*. We will call this agent the *event manager*. The event manager uses the *AED*, the *APD* and the *AIB* as input documents. The event manager can be considered an organizational function that "scans" the application subject area for event occurences. Once the event occurences are detected by the event manager it will determine to what event type the event instance belongs (*event recognizer*, see [29, p.199]). This is done by scanning the AED. Once the event type is determined, the impulse types that contain this event type will be selected. The event manager will now add the event occurence on the *event condition check list* (see figure 5) together with the moment in relative time in which the condition according to the qualified impulse in the event description has to be checked.

EVENT CONDITION CHECK LIST at relative time: t0

Event instance	**condition check on**	**condition type**	**process to be executed**	**at relative time**
(E1(arg1: Piet Jannsen),t-1)	now	CT1	p2	t1
(E1(arg1: Hans Loos),t-1)	now	CT2	p3	t2
(E2(arg2: July),t0)	t1	CT1	p3	t1
(E2(arg2:July,t0)	t1	CT2	p4	t2
(E3(arg4: 23567),t0)	t4	CT1	p2	t5

Fig. 5. Event condition check list[2]

PROCESS EXECUTION LIST at relative time: t1

Event occurence	**process to be executed**	**at moment in relative time**
(E1(arg1: Piet Jannsen),t0)	P2(arg1: Piet Janssen)	now
(E2(arg2: July),t1)	P3(arg2: Augustust)	t2

Fig. 6. The process execution list

The event-condition check list will be evaluated by the *process trigger manager* (see figure 5). The process trigger manager will check the conditions in the impulse at the moment in which the specified moment in time form the impulse will be equal to the current time. If the condition in the impulse is satisfied then the process that should be executed will be added to the *event condition check list* and in addition the trigger time for the conceptual process will be derived from *the condition-process trigger type* as specified in the impulse (type).

Definition 16. The *conceptual event processor* is a prescriptive document that specifies how to evaluate events and conditions.

It should be noted from figure 5 that the condition/process trigger time can be a function of the state of the information base at moment "now", therefore, this expected process execution time should be recorded in this document[3]. The *event condition check list* serves also as an input document for checking the conditions if the relative time changes, e.g. from t_1 to t_2. The processing and evaluating of the input document and checking the event occurence in the subject area will be performed by using the *conceptual event processor* as a prescriptive document. The results of the application of this

[2] An event occurence is referenced by an event instance and the relative moment in time in which the event instance has occured.

[3] This means that the qualification of a condition-process trigger can contain a derivation rule that is defined on fact instances from the application information base.

procedure have to be recorded in order to instantiate the conceptual processes using the *impulse mapper* from the conceptual event description. The event manager needs a document on which to record what process(es) need to be executed in future (relative) time: *The process execution list* (see figure 6). This means that on the *process execution list* only those impulses will be recorded that satisfy the condition on the AIB.

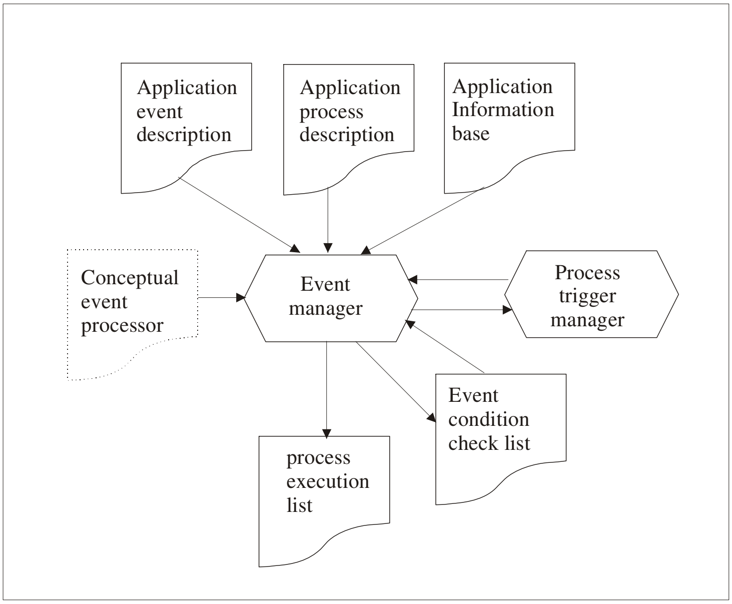


Fig. 7. The event manager subject area

In figure 7 the 'real-life' communication documents that are used and created by the event manager user group in the runtime management UoD are given.

4.2 The Process Manager User Group

The first task of the event manager is to evaluate all events and check wether the impulse-conditions are not violated at the moment in relative time of the event-condition trigger. Secondly the process types will be instantiated when their execution time is due. The second user group within runtime management we will call the *process manager*. This user group will be responsible for the process execution and it will use a *conceptual process processor* that will *instantiate*, *sequence* and *execute* triggered process types. We thereby use the document on which the future executions of conceptual process types are recorded by the event manager : the *process execution list* (see figure 6) as an input. The *process execution list* will show the conceptual processes that are planned to trigger in the future. The *conceptual process processor* is the organizational function that takes care of the execution of multiple processes 'at the same time' and makes it appear that a compound process is executed at a single moment in relative time. Furthermore, the process manager has to be able to sequence conceptual process executions within the same moment of time (e.g. an age can only be derived if the date of birth is known). We will call this function the *process inference manager*. This 'conceptual' manager communicates the proper sequence of conceptual process instantiations for execution by inspecting their *pre- and post conditions***.** The successful execution of a conceptual process will lead to the generation of an instance of a post condition for each fact that is created in a process *Create instance of fact type*. The primary task of the process manager is to generate either an INSERT(fact created) or UPDATE(fact created) request for each fact instance that is created in such a conceptual process instance The secondary task of the process manager is to inspect the *external update request list* and compare it with the *APD*. Only those fact instances can be contained on this external update request list that are created outside the sphere of influence (in an enter process type).

Definition 17. The *conceptual process processor* is a prescriptive document that specifies how to evaluate post- and pre-conditions of conceptual process types and how to sequence the execution of conceptual process at some relative time t.

UPDATE REQUEST LIST at relative time: t2

execution priority	update request	event that caused process	at time
1.	INSERT(arg1: Piet Jansen is kaal)	(E1(arg1: Piet Jannsen),t0)	now
2.	INSERT(arg1: jan koes is niet kaal)	(E4(arg2: July),t1)	now
3.	SELECT(arg1: <person> having salary >1000)		now

Fig. 8. The update request list

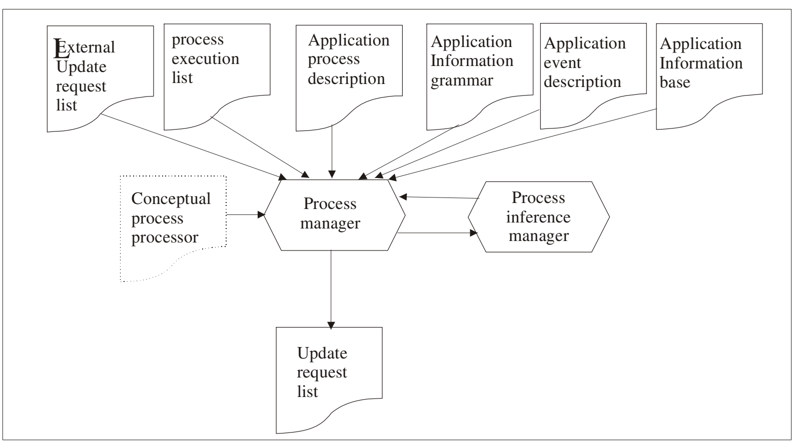


Fig. 9. The process manager subject area

4.3 The Information Base Manager User Group

The third user group within runtime management is the *information base manager* (see figure 11). The information base manager takes the *update request list* (see figure 8) as an input document and will communicate further with another conceptual agent: *the information base update manager.* By checking the *information base before*, the *proposed information base after* and the *application information grammar* of the subject area the information base manager will conclude in a dialogue with the *information base update manager* whether a proposed (compound) update request on the information base will be successfull.

INFORMATION BASE UPDATE REPORT for relative time: t1

Fact instance	**Status**
Jake was born on July 23, 1987	Added
The curent temperature is 23 degrees	Removed
Jim was born on July 29 1988	Rejected

Fig. 10. Information base update report

The *information base update manager* is a function that evaluates proposed information base updates at moment in relative time *t* and the information grammar and the (proposed) information base. The function of the *information base manager* is three-fold. Firstly, The *information base manager* checks whether adding, updating or removing fact instance(s) in (from) the information base is allowed according to the AIG. Secondly, in case this is allowed, (a) fact instance(s) will be added, updated or removed in (from) the information base. The *information base update report* (figure 10) serves as the *proof of acceptance* of the facts that are negotiated or proposed in a conceptual process (or proposed outside the 'sphere of influence') and in addition it informs the relevant environment of the application information system whether *update requests* have been *accepted* or *rejected*.

In line with [17: p.4] we now will give a definition of the *conceptual information processor.*

Definition 18. The *conceptual information processor* is a prescriptive document that specifies how to evaluate that the *AIB* will be in compliance with the *AIG* and how to evaluate the update request list and how to change the information base accordingly.

The messages that are sent whenever a state in the *information base* changes are firstly, recorded in the *information base update report*, secondly these messages are interpreted as potential events, while creating the *AED.* An example of such an information base event is the following*: The fact instance "Jake was born on July 30, 1987" of fact type FT3 is added to the information base.*

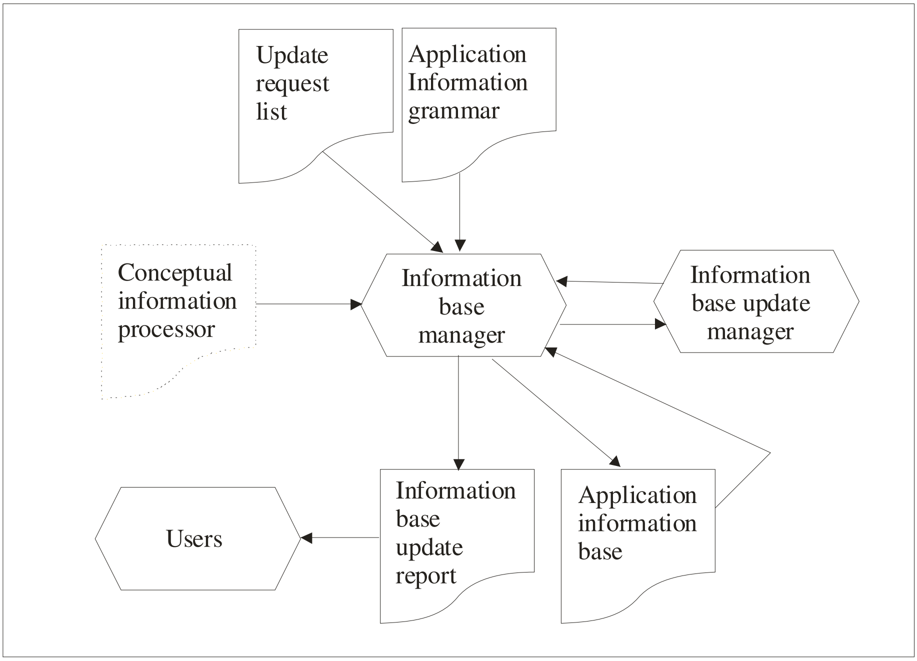


Fig. 11. The information base manager subject area

4.4 Conclusions on Runtime Management

We can now extend the application universe of discourse with the following documents: *event-condition check list*, the *process execution list*, the *update request list* and the *information base update report*. Furthermore, we can integrate the *conceptual information processor* document, the *conceptual process processor* document and the *conceptual event processor* document into the *APD* and the *AED*. We can conclude that the application independent *information meta grammar* is the meta concept in the *information-oriented* perspective, the application independent *information base management processes* is the meta concept in the *process-oriented* perspective and finally the application *independent runtime management* is the meta concept in the *behaviour-oriented* perspective.

5 Business Simulation and Regular Application Domains

We also need to create real-life examples of the reports that we expect from a simulation. These reports will than subsequently be added to the application UoD, and the NLM modeling steps will be applied on this extended UoD. The most essential requirement for the conceptualization of a 'simulation' UoD is that the *runtime management* must be part of the application UoD and in that case it can perform the role of 'simulation executive' [1]. The documents that contain the desired simulation output, can be considered 'real-life' communication documents, for a specific part of a business application. Adding these document to the 'real-life' application UoD, and applying the fact-based modeling methodology for conceptual modeling will result in an extended AIG, in which the simulation fact types are contained including the population constraints. With respect to the APD we can take the process description for the 'real' UoD as a starting point. In order to derive a business simulation application UoD from this 'real' application UoD, the determination process types have to be transformed into derivation process types. This will normally take place by introducing some form of 'stochastic' process, in which a probability distribution, must be selected first. Secondly, the parameters of such a distribution must be set or estimated (e.g. μ (distribution mean) and σ (standard deviation)), see for an example [23,p.659]. Once this has been done, the resulting process can be considered to be derivation processes in the context of the application business simulation UoD by incorporating a random generator. Finally, we need to add an environmental process that generates the 'updates' from the (external) update request list.

In figure 12 we have summarized how some of the elements in the conceptual schema of a general application UoD can be used as a starting point for defining the derivation processes and impulses for the 'simulation' part of the application UoD. It follows from this restriction to derivation process types and impulse mappers, that runtime management can act autonomously in the simulated application UoD. In the next section we will briefly discuss a simplified case study that illustrates the modeling concepts that we have introduced for simulations in this article.

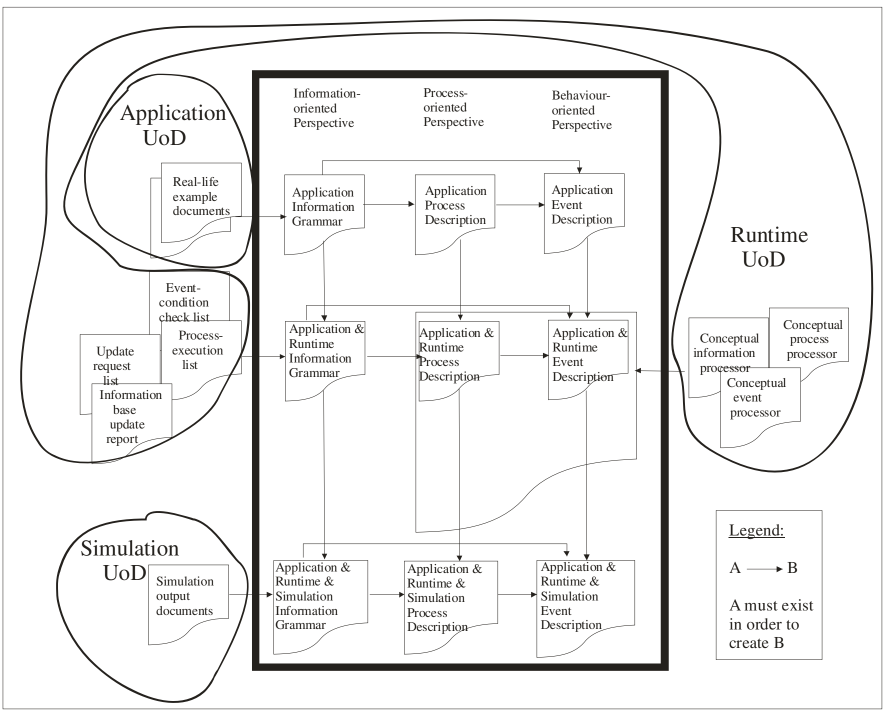


Fig. 12. Relevant model elements for each perspective in 'real-life' versus simulation

5.1 The Citizen Service Case Study

In a city hall of an average city, citizens are able to use the following services: registering a new-born (NB), apply for a passport (PP), apply for citizenship (CS), apply for or renew a driver's license (DL) and ask for a birth-certificate (BC). The way that this service organization operates is as follows. Citizens enter the building and then have to apply for a counter-receipt. In this ' transaction' their time of arrival is recorded. Furthermore, the requested service has to be specified. As a result the customer receives a ticket with the arrival ID (unique on any given day).

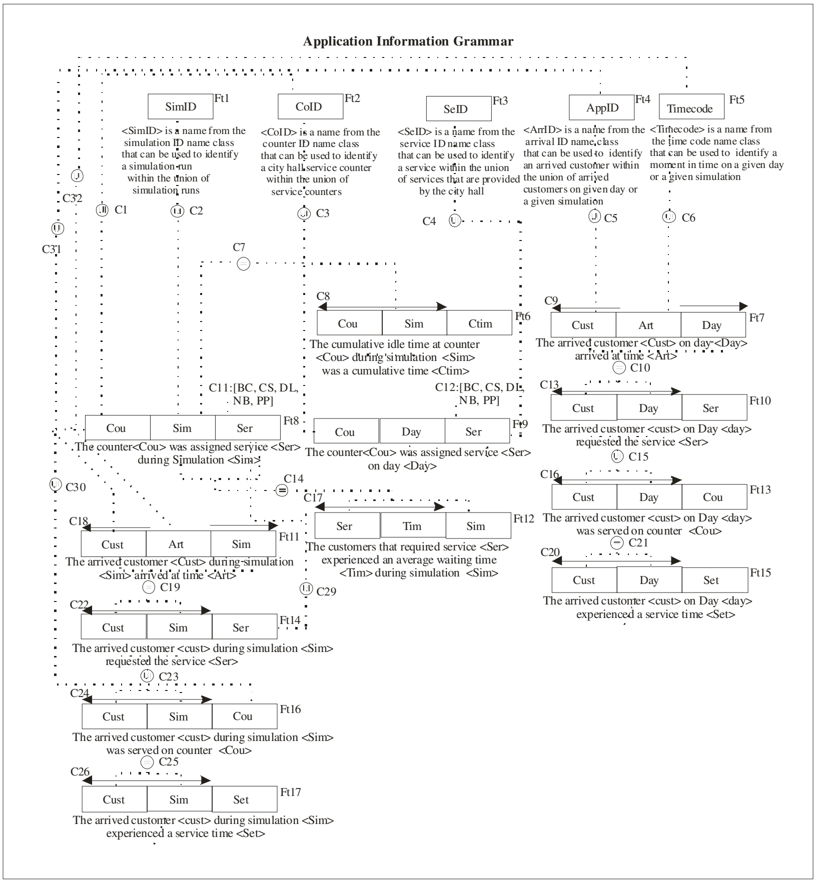


Fig. 13. Partial Application and simulation information grammar

At any time 0, 1, 2, 3 or 4 counters can be operational. In general, one or more services can be performed at any counter, depending upon the experiences of the civil servant. Every time a 'customer' has been processed at a counter, the next in line will be announced on a central display. The central display list the arrival ID plus the number of the counter where this client will be serviced (1, 2, 3 or 4). The AIG, APD and AED for this application subject area are straightforward. In this example we will focus on the simulation requirements. The management of the city hall is interested in the effects of opening/closing one (or more) counters on the (average) waiting times for the clients and the (average) idle time (for a given counter) for the civil servants. The simulation requirements, furthermore are based upon the assumption that customer for every service arrive at the city hall service counter according to a Poisson distribution and that the service times (for a specific service) follow a (negative) exponential distribution [13, p.295-297]. Parts of the resulting AIG, APD and AED for the integration of the 'operational' UoD and the 'simulation' UoD are given in figures 13, 14 and 15.

Pt1 determine service counter for arrived customer (arg1: customer, arg2: service)
IF there exists a FT10.cust where FT10.ser='arg2'
THEN SELECT from FT9.cou where FT9.ser='arg2'
Such that EXPQUELENGTH('FT9.cou') is minimal
FT13.cou=MIN(FT9.cou) where FT13.cust='arg1'
Pt2 determine service counter for simulation customer (arg1: customer, arg2: service)
IF there exists a FT14.cust where FT14.ser='arg2'
THEN SELECT from FT8.cou where FT8.ser='arg2'
Such that EXPQUELENGTH('FT8.cou') is minimal
FT16.cou=MIN(FT8.cou) where FT16.cust='arg1'

Fig. 14. Partial application and simulation process description

From the partial application and simulation process- and event - description in figures 14 and 15 we can see, how the application processes can be used to define the simulation processes.

ON *Et1: customer arrives (arg1: customer, arg2: time, arg3:service)*
DO *Pt1:determine service counter for arrived customer (arg1: customer, arg2: service)*
Where pt1.arg1:=et1.arg1 and pt1.arg2:=et1.arg3
ON *Et2: simulation customer arrives (arg1: customer, arg2: time, arg3:service)*
DO *Pt2:determine service counter for simulation customer (arg1: customer, arg2: service)*
Where pt2.arg1:=et2.arg1 and pt2.arg2:=et2.arg3

Fig. 15. Partial application and simulation event description

6 Conclusion

In this paper we have introduced the fact-based modeling methodology for conceptual modeling and we have shown how it can be used to model not only the 'intentional' communication in an application subject area, but also how it can be used to define the prescriptive and declarative communication documents in the runtime management subject area. Finally, we have shown how the conceptual schema for a simulated application subject area can be incorporated into the 'regular' conceptual documents for an application subject area by defining the 'real-life' communication documents for the simulation UoD and by adding, the simulation derivation process types, and impulses leading to the integrated application information grammar, process description and event description of an application subject area including its simulation requirements. Fact-based modeling turns out to be an approach that allows for a seamless integration between operarional enterprise systems and data on one hand, and the simulation systems and data on the other hand, for the static models (information-oriented perspective) as well as for the behavioural models (process- and behaviour oriented perspectives).

References

1. Ball, P.: Introduction to discrete event simulation. In: Proceedings of the 2nd Dycomans Workshop on: Management Control: Tools in Action, Algarve (1996)
2. Berends, P., Romme, G.: Simulation as a research tool in management studies. European Management Journal 17(6), 576–583 (1999)
3. Bollen, P.: On the applicability of requirements determination methods'. Doctoral Dissertation, University of Groningen (2004)
4. Bollen, P.: Conceptual Process Configurations in Enterprise Knowledge Management Systems. In: Proceedings of 21st ACM Symposium on Applied Computing, SAC 2006, Dijion, France (2006)
5. Bollen, P.: Fact-oriented modeling in the data-, process- and event perspectives. In: Meersman, R., Tari, Z., Herrero, P. (eds.) OTM-WS 2007, Part I. LNCS, vol. 4805, pp. 591–602. Springer, Heidelberg (2007)
6. De, P., Sen, A.: A new methodology for database requirements analysis. MIS Quarterly, 179–193 (September 1984)
7. Floss, P.: Requirements for transitioning business process simulation models to real- time operational systems. In: Andradottir, H.K., Withers, D., Nelson, B. (eds.) Proceedings of the 1997 Winter Simulation Conference, WSC 1997, pp. 1357–1360 (1997)
8. Giaglis, G., Paul, R., Hlupic, V.: Integrating simulation in organizational design studies. International Journal of Information Management 19, 219–236 (1999)
9. Greasley, A.: Using process mapping and business process simulation to support a process-based approach to change in public sector organization. Technovation 26, 95–103 (2006)
10. Gregoriades, A., Karakostas, B.: Unifying business objects and system dynamics as a paradigm for developing decision support systems. Decision Support Systems 37, 307–311 (2004)
11. Halpin, T.: Information Modeling and Relational Databases: from Conceptual Analysis to Logical Design. Morgan-Kaufman, San-Francisco (2008)
12. Hlupic, V., Robinson, S.: Business process modeling and analysis using discrete- event simulation. In: Medeiros, D., Watson, E., Carson, J., Manivannan, M. (eds.) Proceedings of the 1998 Winter Simulation Conference, WSC 1998, pp. 1363–1369 (1998)
13. Julka, N., Lendermann, P., Chong, C.-S., Liow, L.-F.: Analysis and enhancement of planning and scheduling applications in a distributed simulation testbed. In: Ingalls, R., Rossetti, M., Smith, J., Peeters, B. (eds.) Proceedings of the 2004 Winter Simulation Conference, WSC 2004, pp. 1799–1806 (2004)
14. Krajewski, L., Ritzman, L., Malhotra, M.: Operations Management: Processes and Value Chains. Pearson/Prentice-Hall, New Jersey, USA (2007)
15. Lemmens, I., Nijssen, M., Nijssen, G.: A NIAM 2007 conceptual analysis of the ISO and OMG MOF four layer metadata architectures. In: Meersman, R., Tari, Z., Herrero, P. (eds.) OTM-WS 2007, Part I. LNCS, vol. 4805, pp. 613–623. Springer, Heidelberg (2007)

16. Nidumolu, S., Menon, N., Zeigler, B.: Object-oriented business process modeling and simulation: A discrete event system specification framework. Simulation Practice and Theory 6, 533–571 (1998)
17. Nijssen, G.: Databases of the future: design and methodology. Database Journal 12(1), 2–14 (1981)
18. Nijssen, G.: An Axiom and Architecture for Information Systems. In: Falkenberg, E., Lindgreen, P. (eds.) Information System Concepts: An In-Depth Analysis, pp. 157–175. North-Holland, Amsterdam (1989)
19. Nijssen, G., Bijlsma, L.: A conceptual structure of knowledge as a basis for instructional designs. In: ICALT 2006, Kerkrade, the Netherlands (2006)
20. Perera, T., Lyianage, K.: Methodology for rapid identification and collection of input data in the simulation of manufacturing systems. Simulation Practice and Theory 7, 645–656 (2000)
21. Robertson, N., Perera, T.: Automated data collection for simulation? Simulation Practice and Theory 9, 349–364 (2002)
22. Rolland, C.: Database dynamics. Data base, 32–43 (spring 1983)
23. Sadoun, B.: Optimizing the operation of a toll plaza system using simulation: a methodology. Simulation 81(9), 657–664 (2005)
24. Schriber, T., Brunner, T.: Inside discrete-event simulation software; how it works and why it matters. In: Andradottir, S., Healy, K., Withers, D., Nelson, B. (eds.) Proceedings of the 1997 Winter Simulation Conference, WSC 1997, pp. 14–22 (1997)
25. Sol, H.: Simulation in information systems development. Doctoral dissertation, University of Groningen, the Netherlands (1982)
26. Sprague Jr., R.: A Framework for the Development of Decision Support Systems. MIS Quarterly, 1–26 (December 1980)
27. Swaan Arons de, H., Boer, C.A.: Storage and retrieval of discrete-event simulation models. Simulation Practice and Theory 8, 555–576 (2001)
28. de Vreede, G.-J., Verbraeck, A., van Eijck, D.: Integrating the conceptualization and simulation of business processes: A modeling method and arena template. Simulation 79(1), 43–55 (2003)
29. Wieringa, R.J.: Requirements Engineering: Frameworks for Understanding. Wiley, Chichester (1996)

A Framework for the Semantic Representation of Business Processes within Business Organizational Models

Ana Belén Pelegrina, Kawtar Benghazi, María Visitación Hurtado, and Manuel Noguera

Departamento de Lenguajes y Sistemas Informáticos
University of Granada
C/ Periodista Daniel Saucedo Aranda s/n
18071, Granada, Spain
{abpelegrina,benghazi,mhurtado,mnoguera}@ugr.es

Abstract. The modeling of business process is a crucial task in business-based system engineering, from system analysis to simulation. In recent years, several languages to model business processes have been proposed. However, those languages usually lack full integration with other organization structural models, limiting its applicability to real systems. This paper introduces a framework that helps describe business processes and structural organization models in an integrated manner. The framework includes a conceptual model to represent, with enriched semantics, the structure and behavior of an organization, including support for BPMN-based business process models. Several research results point to ontologies as a suitable technology for semantically modeling enterprise systems, from both structural and behavioral perspectives. In this regard, the framework makes use of a set of ontologies in the Web Ontology Language (OWL) [1], which allows implicit knowledge about organizations structure and behavior to be inferred.

Keywords: Business Process Modeling, Business Organizational Models, BMPN, ontologies, OWL.

1 Introduction

"A business process is a persistent unit of work started by a business event [...] driven by business rules that trigger tasks and subprocesses." [2]. As a key activity when designing software for large organizations, Business Process Modeling helps to formalize existing processes and set the focus on needed improvements, facilitate automated, efficient process flow, increase productivity and reduce the number of people involved in the process.

At present, there exist several languages and notations to model business processes, such as the Unified Modeling Language Activity Diagrams [3] (UML AD), Business Process Modeling Notation (BPMN) [4] and Extended Business

J. Barjis (Ed.): EOMAS 2010, LNBIP 63, pp. 79–94, 2010.

Modeling Language (xBML) [5]. However, none of them provides comprehensive integration with configuration models (i.e., organizational structure in terms of groups and roles, relationships between entities, restrictions, etc.). Fairly often it happens that organization policies, such as the granted access to enterprise resources, are defined in structural models on the basis of certain roles defined. Then, business processes are defined in behavioral models where access to organization resources is assumed. In this cases, the lack of alignment between structural and behavioral models may lead to inconsistent business specifications with unforeseen consequences [6].

Business process modeling is not trivial: usually the processes to model are complex and involve a wide range of actors, activities and organizations [7]. Using ontologies as a tool to formalize business processes models may contribute to address that complexity. Furthermore, a formalized model could enable automated inference of model properties. For instance, by including in the ontology a set of restrictions describing when a process is not correctly modeled (e.g., in a business process model, for each event to be caught by an activity, there has to be at least a process that triggers that event) could facilitate the verification process of the models.

AMENITIES [8] is a methodological framework for the study and development of collaborative systems which has been widely used in different research programs from the Spanish Government. In this framework, different views and models are subsequently used by stakeholders (e.g., system architects, system analysts, users, testers, programmers, etc.) in order to capture and analyze the main characteristics of an organization (e.g., system structure and behavior to be supported). AMENITIES extends and makes use of the UML [3]. The models and views proposed in the methodology have been mapped onto formal ontological descriptions in OWL [9]. However, as far as business processes are concerned, UML models and semantics are less expressive than BPMN to represent complex behaviors in business models.

This research work proposes the inclusion in the AMENITIES framework of support for the visual modeling of business processes by using the BPMN notation. This extension is based on the representation of the framework in the OWL language. The use of OWL ontologies offers several advantages like the support for automated reasoning about models. In this regard, the new features of the last version of OWL offer an extended support for advanced logic inference [10]. We provide not only a basis for modeling business processes using BPMN, but also mechanisms to check the correctness of those models. We argue that the integration of BPMN in AMENITIES allows both structural and behavioral models of an organization to be connected which results in a more comprehensive framework for business modeling.

The remainder of the article is organized as follows. In Section 2, we analyze some related pieces of work. Section 3 introduces the set of technologies that serve as a basis for our proposal. Section 4 presents the approach followed in order to implement it. The benefits of using an ontological approach are shown in Section 5. A case study is described in Section 6. Finally, we present the conclusions and ongoing work in Section 7.

2 Related Work

The use of ontologies to model how enterprises organize and plan their business and resources is not a novelty. In fact, the usefulness of an ontological approach to model enterprise systems has been widely advocated, such as in [11] and [12]. Going further with this approach, Dietz and Hoogervorst [13] define *Enterprise Ontology* as "*[...] the implementation independent essence of an enterprise, understood from a holistic systemic point of view.*". They propose the definition of system ontologies based on the Ψ-theory (Performance in Social Interaction), providing a sound basis for *Enterprise Engineering* including organizational and behavioral aspects.

As for the suitability of BPMN to model business processes, Wahl and Sindre [14] perform an analytical evaluation of BPMN using the Semiotic Quality Framework, concluding that BPMN particularly exhibits a good performance in terms of comprehensibility and suitability for the domain of business process modeling. Wohed et al. [15] evaluate BPMN using the Workflow Patterns identified by van der Aalst [16] as an evaluation framework. The results of the evaluation show that BPMN lacks a proper representation for data and resource perspectives. Other piece of work that evaluates BPMN is [17], which compares BPMN to UML AD and the Extended Enterprise Modeling Language (EEML) [18] using a case study as a basis; BPMN attains the best results in all the categories but one: domain suitability.

Ghidini et al. [19] present a mapping from BPMN to OWL. However, this mapping does not support some n-ary relations defined in the BPMN metamodel, it is not included within other organizational model and does not offer any extended support for the reasoning process about the correctness of the models.

3 Background

This section introduces the technologies that underpin a semantic framework for organization modeling. First of all, BPMN is briefly described. Then, ontologies, OWL and the Semantic Web Rule Language (SWRL) [20] are explained. Finally, the AMENITIES conceptual framework is presented.

3.1 Business Process Model Notation

BPMN [4] aims to offer a notation that is easily understandable by all the stakeholders and analysts while being able to represent complex process semantics. BPMN notation is instantiated in Business Process Diagrams (BPD), which are based on a flowcharting technique similar to that of activity diagrams from UML. BPDs consist of graphs where the nodes are the activities (units of work) and the connections between them define the development of a workflow. BPMN elements are divided in two sets: *Core Elements*, which include the basic elements to define a business process model and *Extended Elements*, which extend the core elements and provide richer semantics. These elements are classified in four categories: (1) *Flow Objects* are the main elements that define the behavior of a Business Process; (2) *Connection Objects* connect *Flow Objects*; (3) *Swimlanes* group the basic elements by organizations/roles and (4) *Artifacts* provide additional information about the process.

3.2 Ontologies

In Computer Science, an ontology can be defined as *"an explicit specification of a conceptualization"* [21]; ontologies allow a common vocabulary for a domain to be defined, including concepts and objects, their properties, relationships between them and restrictions that apply to these concepts and objects. Many languages for the specification of ontologies are based on some type of logic (e.g., Description Logics [22], first order logic, etc.), which permits to reason about the knowledge expressed in the ontology. Business process modeling may be improved with the use of ontologies as a way to formalize them, taking advantage of the modularity, interoperability and reasoning capabilities of ontologies.

An area where the use of ontologies is more widespread is the Semantic Web [23]. One of the most used languages in the Semantic Web is the Web Ontology Language (OWL), endorsed by the W3C. At present, the use of OWL in environments other than that of the Semantic Web is steadily increasing. OWL is based on Description Logics and enables automated reasoning procedures over OWL ontologies descriptions. The current version, OWL 2, includes some new characteristics that avoid some of the limitations of the previous version of the language while preserving decidability. These new features include, among others: increased expressive power for properties, extended support for datatypes and simple meta-modeling capabilities. Closely related to OWL is SWRL, a rule language based on a combination of the OWL and Rule Markup Language (RuleML) [24]. It extends the set of OWL axioms to include Horn-like rules. It thus enables rules to be combined with an OWL knowledge base. SWRL aims to increase the expressivity of OWL ontologies.

3.3 AMENITIES

AMENITIES is a methodological framework intended to address the organizational and operational modeling of cooperative systems (e.g., an enterprise). This framework makes use of different UML-based models, which could be structural o behavioral models, in order to enable system modeling by stakeholders, (e.g., business analysts, users, developers, etc.). However, UML lacks of formal semantics, making difficult to automatically carry out verifications on the models. As part of the work developed in [25], AMENITIES has been formalized as an OWL ontology, providing clearly-defined semantics and enabling the automatic verification of models.

The framework also includes a conceptual model that serves as a reference model to describe the structure and behavior of an organization. This model includes the main concepts that are present in enterprise systems, as well as the relationships between those components. This conceptual model incorporates concepts such as

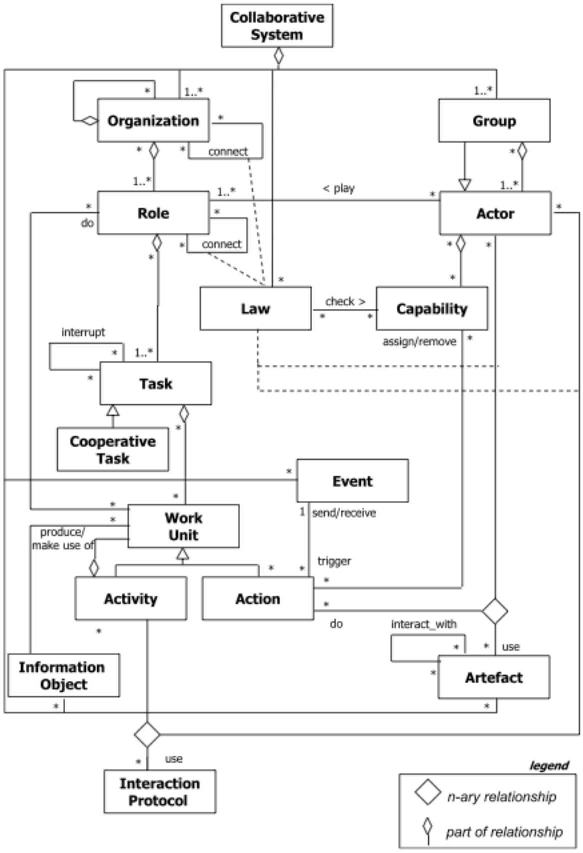


Fig. 1. AMENITIES conceptual model for collaborative systems

actors, roles, tasks and communication protocols. The conceptual model is shown in the class diagram depicted in Fig. 1.

Despite its initial orientation towards collaborative systems modeling, AMENITIES may be used to model business processes, employing some of the framework-defined concepts as elements of the business process model. Specifically, it makes use of UML AD as a way to model business processes.

As for the ontological representation of UML AD, several issues arise when attempting to describe them in OWL. For instance, OWL does not offer direct support for modeling sequences. Consequently, as business processes are usually described in terms of lists of activities arranged in some manner, it is necessary to use some workarounds to overcome these limitations [26]. The adopted solution is to provide a set of constructs to rule over the control flow in a activity, (e.g., initiate, finalize, join, fork, merge, choose, etc.), applying a set of patterns to circumvent OWL limitations. In Fig. 2 is shown a version of this extension.

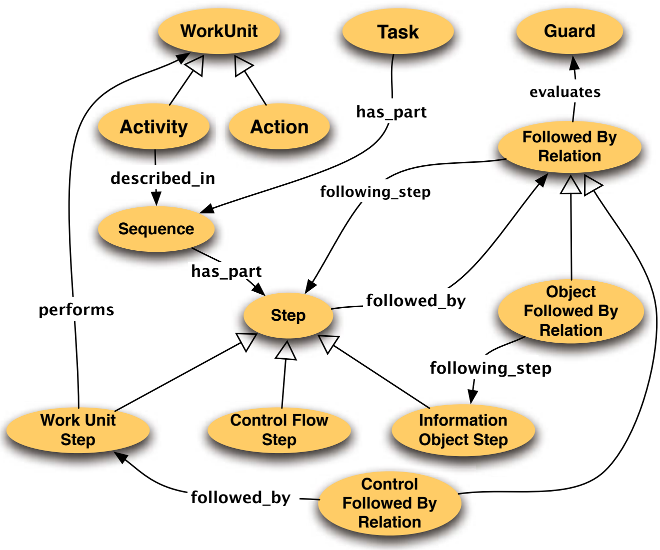


Fig. 2. AMENITIES extension for UML AD

4 Domain Model for BPMN within AMENITIES

UML AD are commonly used in order to model business processes. One of the strengths of UML AD it is the wide use of UML in the software industry and the high number of tools available. However, business process modeling usually involves non-technical stakeholders, so that the use of a technical language may not be advisable. On the contrary, BPMN intends to appeal to a broader audience than that of UML AD, especially for stakeholders without a computer/technical background. As a consequence, BPMN offers a simpler syntax and provides a wide set of constructs and abstractions over some recurrent business process patterns (e.g., transactions, predefined typed events and compensation activities), enabling richer semantics.

UML AD and BPMN present very similar expressive capabilities [27] when modeling common patterns describing the behavior of business processes, thus, we may expect that some of the concepts used to model UML AD can be reused to model their BPMN counterparts. Despite the similarities between UML AD and BPMN, some elements in BPMN have not a counterpart in UML AD. Therefore, for the non supported BPMN elements, it has been necessary to add new classes and properties to the AMENITIES ontology so as to provide a complete support for BPMN. In this respect, flow elements are modeled as relations (object properties in OWL jargon); the rest of BPMN elements, such as events, process and informations objects, are modeled as concepts (i.e. OWL classes). Fig. 3 depicts a conceptual map for the BPMN extension of AMENITIES. In Table 1 the proposed extension for BPMN elements is shown.

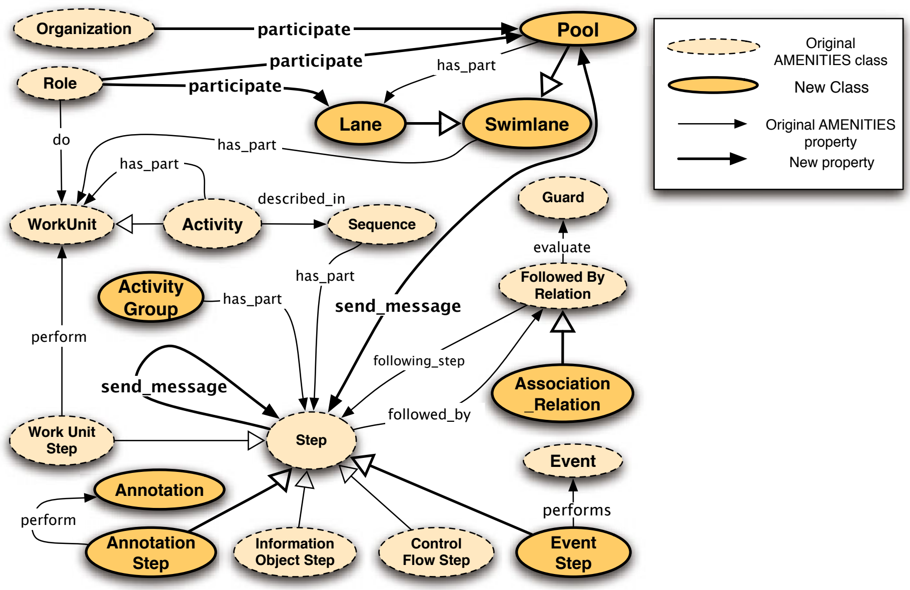


Fig. 3. AMENITIES extension including BPMN *Core Elements*

In some cases, the mapping involves a set of entities, instead of a single class or property. For example, BPMN activities are represented in the extension using two classes: *WorkUnit*, which represents a reusable unit of work and a *Work_ Unit_ Step* which represents a work unit actually performed within a business process, i.e., the work unit steps are instances of an "*abstract*" work unit (see Fig. 3 and Table 1). This representation enables the separation between a process and its actual execution in a workflow. Otherwise, it would not be possible to reference the same process in two places in a business process diagram.

5 Exploiting the Ontological Model

OWL allows the inference of implicit knowledge in an ontology, by means of automated reasoners. This inference may be useful to detect inconsistencies and errors in the business process model. In order to support error detection, we have defined a set of classes, properties and rules in the ontology. The latest version of OWL includes a set of new features that enables further reasoning capabilities on OWL ontologies. In this section, we will present some scenarios where the reasoning capabilities of OWL can be applied in order to facilitate the business process model using AMENITIES.

Although the examples shown in this section are presented using only an OWL ontology IDE, Protégé [28], a graphical tool could be built, using the framework as a reference, in order to enable visual edition of the models and automated reasoning about their correctness.

Table 1. Summary of the BPMN elements and their equivalent representation in the AMENITIES ontology. New classes and properties are bolded.

BPMN Elements	OWL Ontology for AMENITIES concepts
Event	*Event* and *Event_Step* classes (See Section 5 for a further explanation)
Flow Dimension	**StartEvent, IntermediatedEvent, BoundaryEvent** and **EndEvent** classes
Type Dimension	**EventDefinition** class and their subclasses and **has_event_definition** property
Trigger	**CatchedEvent** and **ThrowedEvent** classes
Activity	*WorkUnit* and *Work_Unit_Step* classes.
Process/Sup-Process (non-atomic)	*Activity* and *Activity_Step* classes
Task (atomic)	*Action* and *Action_Step* classes
Transaction	**TransactionalActivity** class
Activity Looping	**LoopingActivity** class
Multiple Instances	**MultipleActivity** class
Gateway	*Control_Flow_Step* class
Exclusive Data-Based	*Decision_Step* and *Merge* classes
Exclusive Event-Based	**Event_Decision_Step** and *Merge* classes
Parallel	*Fork_Step* and *Join_Step* classes
Complex	**Complex_Decision_Step** and *Merge* classes
Inclusive	**Inclusive_Decision_Step** and *Merge* classes
Sequence Flow	*Step,* and *Control_Followed_By_Relation* classes and *following_step* and *followed_by* properties
Uncontrolled Flow	*Step,* and *Control_Followed_By_Relation* classes and *following_step* and *followed_by* properties
Conditional Flow	*Guard,* and *Control_Followed_By_Relation* classes and *following_step, evaluates* and *followed_by* properties
Default Flow	**Default Followed_By_Relation**, *Step, Guard,* and *Control_Followed_By_Relation* classes and *following_step, evaluates* and *followed_by* properties
Exceptional Flow	**Exceptional_Followed_By_Relation** class and *following_step* and *followed_by* properties
Association	**Association_Relation** class and *following_step* and *followed_by* properties
Compensation Association	**Association_Compensation_Relation** class and *following_step* and *followed_by* properties
Message Flow	**send_message** property
Swimlane	**Swimlane** and *Work_Unit* classes and *has_part* property
Lane	**Lane**, and *Role* classes and **participate** property
Pool	**Pool**, *Role*, *Organization* and*Lane* classes
Artifact	-
Data Object	*InformationObject* and *InformationObjectStep* classes
Group	**ActivityGroup** and *Step* classes, and the properties *has_part* and *part_of*
Annotation	**AnnotationStep** and **Annotation** classes, which represents an Annotation

5.1 BPMN Event Restrictions

In BPMN, events are defines as "*something that happens during the course of a business process*", having a effect on the flow of the process [4] . BPMN events have a flow dimension, depending on the time the events appear in the sequence

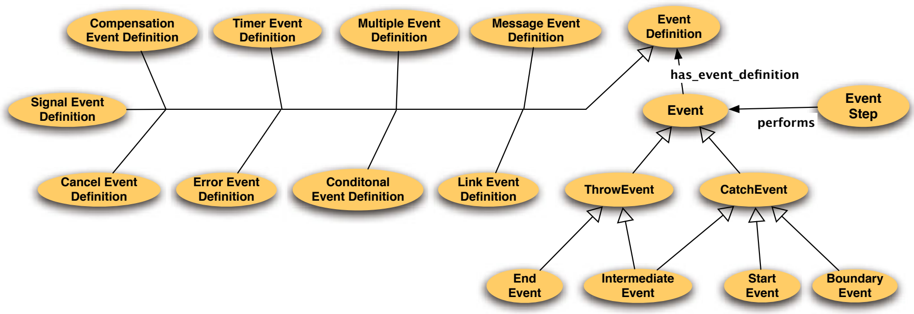


Fig. 4. Events sub-ontology within AMENITIES

flow: *Start Events* (initiate a process/subprocess), *Intermediate Events* (the event occurs in the middle of the process) and *End Events* (finish a process/subprocess). BPMN events can also be associated with a definition (i.e., a type) of event (message, timer, error, etc.). Finally, BPMN events can be thrown (the event is triggered in the process) or caught (the event is captured by the process). In Fig. 4 is represented the event sub-ontology.

As it may be expected, BPMN places some restrictions on the events dimensions (e.g., "*an start event cannot be throwed, only catched*"). In order to comply with these restrictions, the proposed extension includes a set of OWL restrictions are applied to the concepts defined in the ontology, allowing the detection of incorrectly defined events, according to the BPMN event model. As an example, some definitions and restrictions associated with this event model are defined below. In the same manner that these restrictions, the rest of restrictions associated with BPMN events have been defined.

Restriction 1. This restriction reflects the allowed types for *Start Events* in BPMN. This types are: *Message, Timer, Conditional, Signal* and *Multiple.*

```
StartEvent has_event_definition only
    (Message_Event_Definition  or Timer_Event_Definition or
    Conditional_Event_Definition or Signal_Event_Definition or
    Multiple_Event_Definition or Parallel_Event_Definition)
```

Restriction 2. This restriction reflects the allowed types for *Intermediate Events* in BPMN. The only invalid type for an intermediate event is *Terminate.*

```
IntermediateEvent has_event_definition only not Terminate_Event_Definition
```

As an example of the applicability of this restriction, we present the following scenario, as shown in Fig. 5: an *Intermediate Event* has assigned a *Terminate*

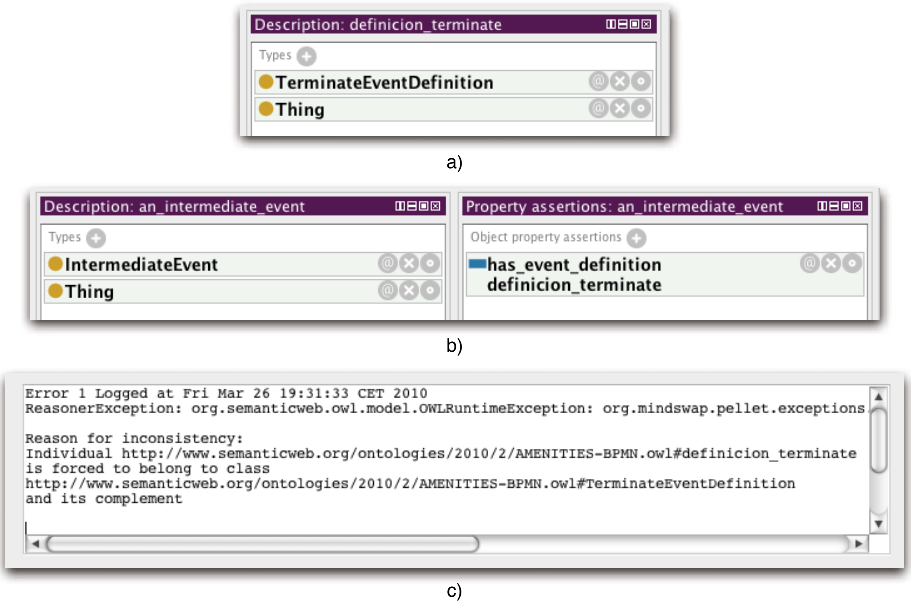


Fig. 5. Detection of an error in the declaration of an *IntermediateEvent*. a) A type of event is defined. b) An intermediate event is declared of type Terminate. b) The reasoner detects the inconsistency and raises an exception.

Definition, clearly, this situation is not valid in BPMN. When attempting to validate this scenario, an error is raised.

5.2 SWRL Rules

Using SWRL rules, we can expand OWL reasoning expressiveness and provide new ways to detect more inconsistencies in a BPD. As examples of the utility of SWRL rules in order to detect possible errors, we present in this section two SWRL rules: one that detects when a message is not correctly exchanged and another that checks if a compensation activity is properly used. In the same manner, other restrictions can be modeled similarly; for example, an *Activity_ Step* cannot be isolated from the flow (i.e., no connected with any other step) and an *EndEvent* cannot be followed by another step.

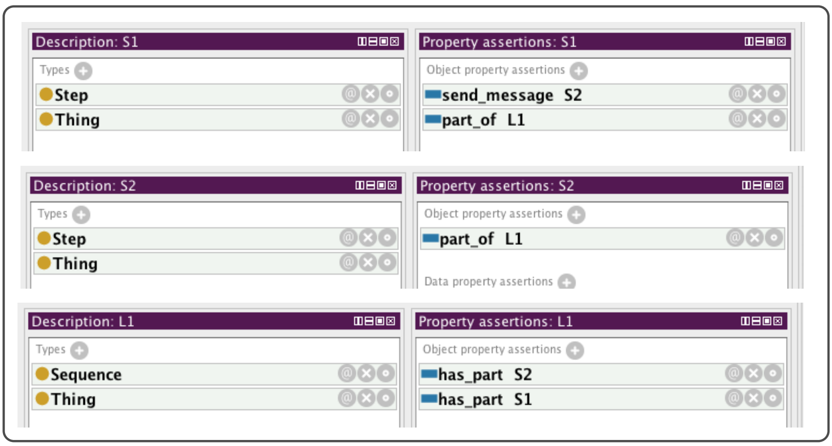
Rule 1. According to BPMN, messages can be only exchanged between activities belonging to different sequences (i.e., an activity cannot send a message to a previous o succeeding steps in its sequence). This rule detects if the activity s1 is sending a message to activity s2, belonging both to the same process. The reasoning process about an simple example that triggers this rule, using Protégé and Pellet [29] (an OWL reasoner), is shown in Fig. 6.

```
Step(?s1) ^  Step(?s2) ^ different(?s1, ?s2) ^ send_message(?s1, ?s2) ^
Sequence(?l1)  ^ Sequence(?l2) ^ has_part(?l1, ?s1)  ^
has_part(?l2, ?s2)  ^ sameAs(?l1,?l2)
                                        -> BadMessengerStep(?s1)
```

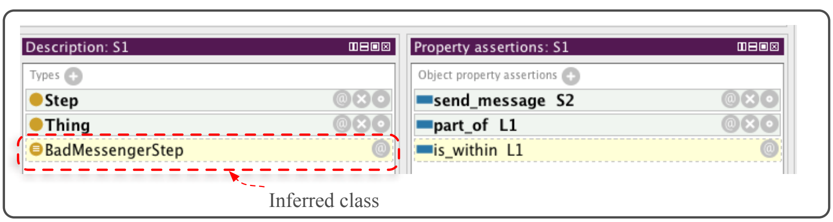
Rule 2. These rules check if there is a sequence flow between a compensation activity and another element (i.e., the compensation activity is followed by other step), which is not valid according to BPMN specification. The firs rule identifies compensations steps and the second identify compensation steps followed by other steps.

```
Step(?s) ^ Association_Compensation_Relation(?r) ^ following_step(?r, ?s)
 -> Compensation_Step(?s)

Compensation_Step(?cs) ^ followed_by(?cs, ?x)
                               -> Malformed_Compensation_Step(?cs)
```



a)



b)

Fig. 6. Result of the application of Rule 1 to a dummy example, using Protégé 4. a) shows the declaration of the sequence L1 and two step, S1 and S2, being both steps part of L1. b) Depicts how the error of S1 sending a message to S2 is detected using an OWL reasoner.

6 Case Study

As a means of showing the applicability of our proposal, we present a case study. This case study is based on a simplified airline tickets booking process. The roles involved in the process are two: the customer that desires to book a flight and the reservation agent, which performs the reservation. A BPD that attempts to model this process is depicted in Fig. 7.

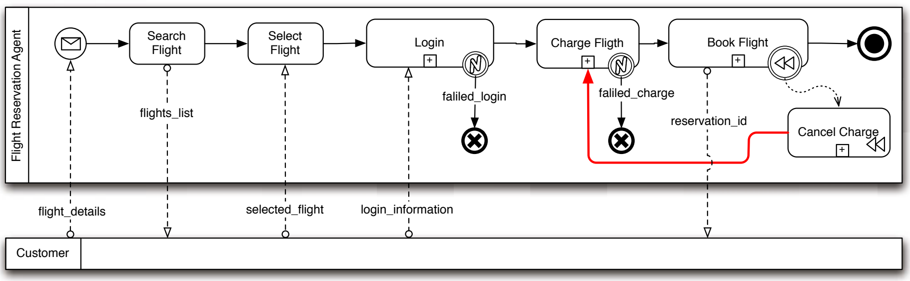


Fig. 7. Business process model for a simple booking process of an airplane ticket using BPMN

The process consist in the following steps:

1. The customer provides the details of the desired flight (e.g., departure city, date, roundtrip, etc.).
2. The reservation agent searches for flights that match user's desires and returns a list of flights to the customer.
3. The customer selects a flight from the list.
4. The customer logs in[1]. If this action fails, the process finishes.
5. The reservation agent charges the customer the amount of the flight. If the charge action fails, the process finishes.
6. The reservation agent books the flight and sends the reservation info to the customer. If an error occurs during this sub-process, the reservation is cancelled and the money is returned.

In order to enable the verification of the business process, it has been formalized into an OWL ontology using the mapping proposed in Section 4. As shown in the BPD depicted in Fig. 7, one error has been made in the modeling: the compensation activity associated with the task Book Flight is followed by another step, which is not valid as specified in BPMN standard. The OWL entities involved in the error are highlighted in Fig. 8.a). On the basis of this ontological description and the restrictions included in the framework, an automated reasoner is able to detect the error applying the rules presented in section 5.2, as shown in Fig. 8.b).

[1] If the user is not registered in the system, he can create a new account, providing his/her personal information and payment data (e.g., credit card number).

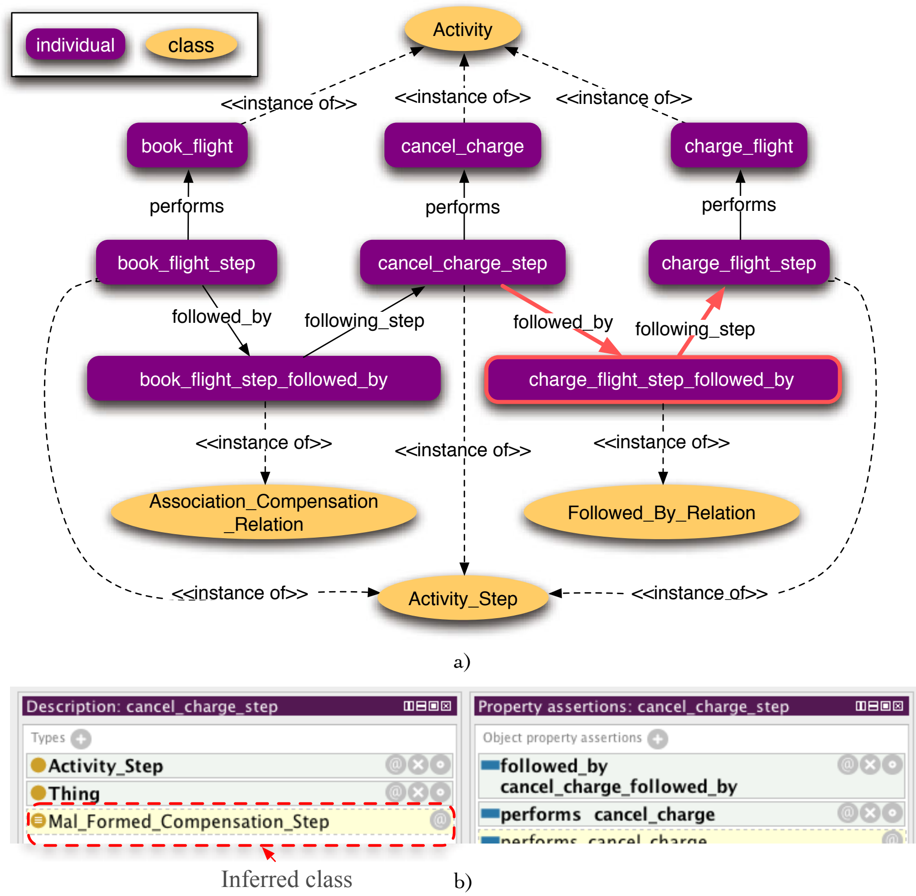


Fig. 8. Formalization of the case study using the proposed framework. a) Conceptual map of the steps *book_ flight_ step*, *cancel_ charge_ step* and *charge_ flight_ step*. b) Error detected after the inference process: the compensation step is set as a *Malformed_ Compensation_ Step* instance.

7 Conclusions and Future Work

AMENITIES is a conceptual framework to support the modeling of organizations, their constituent components and the relationships between them. In this paper, we present an extension to AMENITIES ontological representation so as to enable the semantic description of business processes using BPMN. The starting point is a mapping between the elements in the BPMN notation and concepts and relationships in the AMENITIES conceptual framework. This extension also adds to AMENITIES a set of new classes and properties, enabling the detection of complex relationships between processes, and even errors in their specification. With this underlying ontological description, business process modeling by using BPMN can be improved on the basis of reasoning capabilities of OWL ontologies and the concepts and relationships included in AMENITIES.

The proposal harnesses the power of both types of models: a graphic modeling language for business process models (visual, easy to use and understand) and a conceptual framework for business process and organization modeling with a formal grounding (reasoning, shared vocabulary, verification).

As future work, we plan to incorporate the new features in BPMN 2.0 [30], which is currently in Beta version, as the new types of diagrams (e.g., choreography diagrams) and the full integration with Web Services BPMN (WSBPMN) [31]. We also plan to extend the framework in order to include new concepts and relationships that would allow to infer more complex interdependencies and enrich the semantics of the organizational model. Finally, we plan to develop a tool supporting the proposed framework. This tool will enable the visual modeling and verification of business processes in the BPMN notation. The first version of this tool is currently under development as a Protégé 4 plug-in[2].

Acknowledgment. This research has been funded by the Spanish Government's Ministry of Science and Innovation, via the projects TIN2008-05995/TSI and TIN2007- 60199.

References

1. Motik, B., Patel-Schneider, P.F., Parsia, B.: OWL 2 Web Ontology Language Structural Specification and Functional-Style Syntax. In: W3C OWL Working Group (October 2009)
2. Dayal, U., Hsu, M., Ladin, R.: Business process coordination: State of the art, trends, and open issues. In: 27th International Conference on Very Large Data Bases, pp. 3–13 (2001)
3. Baisley, D., Björkander, M., Bock, C., Cook, S., Desfray, P., Dykman, N., Ek, A., Frankel, D., Gery, E., Haugen, Ø., Iyengar, S., Kobryn, C., Møller-Pedersen, B., Odell, J., Övergaard, G., Palmkvist, K., Ramackers, G., Rumbaugh, J., Selic, B., Weigert, T., Williams, L.: OMG Unified Modeling Language (OMG UML), Superstructure v 2.2. Object Managment Group (OMG) (February 2009)
4. OMG: Business Process Model and Notation (BPMN) v1.2. Object Managment Group (OMG) (2008)
5. Tyler, C.G., Baker, S.R.: Business Genetics. John Wiley and Sons, Chichester (2007)
6. Szymczak, M., Frackowiak, G., Ganzha, M., Gawinecki, M., Paprzycki, M., Park, M.W.: Resource management, pp. 458–462 (November 2007)
7. Barjis, J.: The importance of business process modeling in software systems design. Science of Computer Programming 71(1), 73–87 (2008)
8. Garrido, J., Gea, M., Rodríguez, M.: Requirements engineering in cooperative systems. In: Requirements Engineering for Sociotechnical Systems, pp. 226–244 (2005)

[2] A preliminary version of the tool (supporting only UML AD within AMENITIES) is available at http://code.google.com/p/violet-owl.

9. Noguera, M., Hurtado, M.V., Rodríguez, M.L., Chung, L., Garrido, J.L.: Ontology-driven analysis of uml-based collaborative processes using owl-dl and cpn. Science of Computer Programming 75(8), 726–760 (2009)
10. Golbreich, C., Wallace, E.K.: OWL 2 Web Ontology Language New Features and Rationale. In: W3C OWL Working Group. (2009)
11. Hepp, M., Roman, D.: An ontology framework for semantic business process management. In: Proceedings of Wirtschaftsinformatik, vol. 2007 (2007)
12. Dietz, J.L.G.: System ontology and its role in software development. In: Proceedings of the CAiSE 2005 workshop Interop-EMOI (2005)
13. Dietz, J.L.G., Hoogervorst, J.A.P.: Enterprise ontology in enterprise engineering. In: SAC 2008: Proceedings of the, ACM Symposium on Applied Computing, pp. 572–579. ACM, New York(2008)
14. Wahl, T., Sindre, G.: An analytical evaluation of bpmn using a semiotic quality framework. In: Castro, J., Teniente, E.(eds.): Proceedings of the CAiSE 2005 Workshops, vol. 1, pp. 533–544 (2005)
15. Wohed, P., van der Aalst, W.M., Dumas, M., ter Hofstede, A.H., Russell, N.: On the suitability of bpmn for business process modelling. In: Dustdar, S., Fiadeiro, J.L., Sheth, A.P. (eds.) BPM 2006. LNCS, vol. 4102, pp. 161–176. Springer, Heidelberg (2006)
16. van der Aalst, W., ter Hofstede, A., Kiepuszewski, B., Barros, A.: Workflow patterns. Distributed and Parallel Databases 14(1), 5–51 (2003)
17. Nysetvold, A., Krogstie, J.: Assessing business process modeling languages using a generic quality framework. In: Castro, J., Teniente, E. (eds.): Proceedings of the CAiSE 2005 Workshops, vol. 1, pp. 545–556 (2005)
18. Krogstie, J.: Using eeml for combined goal and process oriented modeling: A case study. In: Proceedings of EMMSAD 2008 (2008)
19. Ghidini, C., Rospocher, M., Serafini, L.: A formalisation of bpmn in description logics. Technical report (2008)
20. Horrocks, I., Patel-Schneider, P.F., Boley, H., Tabet, S., Grosof, B., Dean, M.: Swrl: A semantic web rule language combining owl and ruleml. Technical report, W3C Member Submission (2004)
21. Gruber, T.: A translation approach to portable ontology specifications. Knowledge Acquisition (January 1993)
22. Baader, F., Calvanese, D., McGuinness, D.L., Nardi, D.: The Description Logic Handbook: Theory, Implementation and Applications. Cambridge University Press, Cambridge (2003)
23. Berners-Lee, T., Hendler, J., Lassila, O.: The semantic web. Scientific American (May 1998)
24. Boley, H.: The rule markup language: Rdf-xml data model, xml schema hierarchy, and xsl transformations. In: Bartenstein, O., Geske, U., Hannebauer, M., Yoshie, O. (eds.) INAP 2001. LNCS (LNAI), vol. 2543, pp. 5–22. Springer, Heidelberg (2003)
25. Noguera, M.: Modeling and Analysis of CSCW Systems: an Ontology Driven Engineering Approach (Modelado y Análisis de Sistemas CSCW Siguiendo un Enfoque de Ingeniería Dirigida por Ontologías). PhD thesis, Universidad de Granada (May 2009)
26. W3C: Defining n-ary relations on the semantic web. W3C Working Group Note (April 2006)

27. White, S.A.: Process modeling notations and workflow patterns. BPTrends (2004)
28. Knublauch, H., Fergerson, R., Noy, N., Musen, M.: The protege OWL plugin: An open development environment for semantic web applications. In: McIlraith, S.A., Plexousakis, D., van Harmelen, F. (eds.) ISWC 2004. LNCS, vol. 3298, pp. 229–243. Springer, Heidelberg (2004)
29. Parsia, B., Sirin, E.: Pellet: An owl dl reasoner. In: Proceedings of the International Workshop on Description Logics (2003, 2004)
30. White, S.A.: Business process modeling notation (bpmn), v. 2.0 beta. Technical report, Object Management Group (September 2009)
31. Barreto, C., Bullard, V., Erl, T., Jordan, D., Kand, K., König, D., Moser, S., Stout, R., Ten-Hove, R., Trickovic, I., van der Rijn, D., Yiu, A.: Web services business process execution language. Technical report, OASIS (2007)

ERP Implementation Strategies: The Importance of Process Modeling and Analysis

Angel Díaz[1], Oswaldo Lorenzo[1], and Björn Claes[2]

[1] IE Business School, Mª Molina 11-15, Madrid 28006, Spain
{angel.diaz,oswaldo.lorenzo}@ie.edu
[2] Cranfield School of Management, Cranfield, Bedford, England MK43 0AL
bjorn.claes@cranfield.ac.uk

Abstract. Despite the existence of a vast body of research and knowledge related to ERP implementation, the analysis of environmental aspects (e.g. organizational context, existing business strategies, and implementation strategies) that influence the implementation success remains under researched. Hence, the objective of this study was to gain insight of the relation between the ex-ante business variables (i.e. business characteristics identified prior to the implementation), the ERP implementation strategies, and the perceived final contribution. Using survey data obtained from 91 respondents out of a sample population of 549 SAP users in Spain and Latin America (a net response rate of 16.6%) we find a significant correlation between specific business strategies and particular ERP implementation strategies. In addition we observe that process mapping and process-based implementation strategies better explain implementation results than other strategies such as the use of consultants and previous experience.

Keywords: ERP implementation, process modeling.

1 Introduction

Globalization, market volatility, economic uncertainty and shorter product life cycles are all part of today's business reality in which companies need to compete. The gradual emergence of new information and communication technologies, requires that organizations make significant investments in the efficient implementation of integrated management systems, known by the acronym ERP (Enterprise Resources Planning). These tools are essential a crucial facilitator for the integration of information and a key requirement for achieving greater efficiency in the use of resources, as well as greater flexibility and speed of response [1], [2], [3].

Both academic and practitioner studies coincide in their efforts to identify relations between ERP implementation practices and the implementation results [4], [5], [6]. The vast majority of the studies focus on the performance indicators related to the implementation process that impact the business results. The analysis of environmental aspects that influence the implementation success, such as the organizational context, existing business strategies, and the implementation strategies remains under researched.

J. Barjis (Ed.): EOMAS 2010, LNBIP 63, pp. 95–112, 2010.

The study reported in this paper addresses this gap in the literature. Using cross sectional survey data drawn from a sample population of 549 SAP users originating from Latin America and Spain this study sets out to explore relations between ex-ante business variables (i.e. business characteristics identified prior to the implementation), implementation strategies, and the perceived contribution of the implementation.

This paper is structured as follows: in the following section we detail the conceptual model that guided our exploration, we then elaborate on the data that we collected to test the conceptual model and the methodology we used to analyze this data. This section is succeeded by the analysis of the results and a discussion of the findings. We close the paper with some remarks on the limitations of the study and some suggestions for further research.

2 Model and Variables

The objective of the study reported henceforth was to gain insight in the implementation processes through an exploration of the relation between the ex-ante business variables, ERP implementation strategies, and the perceived implementation contributions to business results. To guide our research we developed the conceptual model sketched below. In this model, we posit a relationship between the strategies that companies device to implement ERPs and the ex-ante business variables on one side and the success of the implementation on the other side. We expected both relations to be moderated by one or more contextual factors.

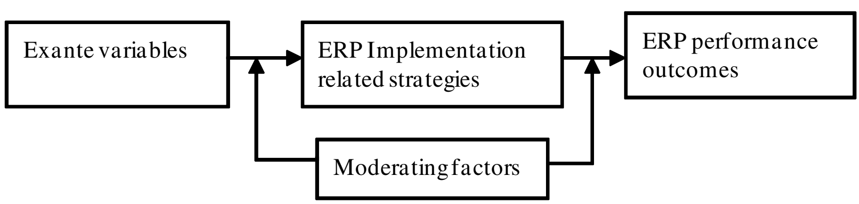


Fig. 1. The conceptual research model

2.1 Ex-ante Business Variables

The ex-ante business variables refer to the antecedents of ERP implementation and include factors such as the dominant competitive strategy of the organization, any prior experience in transformation or system implementation projects, the extent to which upper management participate in these projects, and the process orientation of the organization. For a detailed discussion of these antecedents and how they relate to each other we refer to Lorenzo et al. [7]. In the subsequent description, we will limit ourselves to a brief description of each of the components of the model.

Business Orientation. Following Porter's [8] categorization, the dominant business strategies are divided into three main, non-exclusive approaches (non-exclusive because different product lines can be organized along different strategies within the same firm). Categories include strategies that strive to have the lowest operational costs, strategies aimed at having products or services perceived to be unique, and strategies that focus on a specific market segment. Because the last two strategies were found to be highly correlated[1] a binary split of the main business strategy is henceforth considered: price strategy and differentiation strategy.

Previously Acquired Capabilities. Specific individual and organizational capabilities can affect the ERP performance [9], [10]. Parr, Shanks and Drake [11] argue that organizational members absorb ERP knowledge more effectively if they have prior knowledge about ERP systems. This more effective absorption allows organizational members to take more advantage of the ERP benefits after the implementation.

Coordination mechanisms. Coordination refers to the management of dependencies among activities [12]. One of the benefits of the ERP system after adoption is the improvement of coordination. The ex ante factor that is key for realizing the benefits from coordination is the level of previously established coordination mechanisms within the organization. Process orientation, top management involvement, team development, and information systems are examples of such coordination mechanisms in organizations.

Organizational and Market context. Some research efforts on ERP systems have been dedicated to gaining a better understanding of the influence that contextual factors have on the development and/or the post implementation results of an ERP implementation. Mabert, Soni and Venkataramaran [13] studied this from the organizational size perspective and found that the implementation strategies for ERPs varied in organizations of different sizes. They also found that the benefits realized by the implemented ERPs differed by company size. Everdingen, Van Hillegersberg and Waarts [14] argue that ERP success in small and medium enterprises relies more on shorter implementation times than it does in larger companies. Other scholars have investigated the ERP phenomenon in specific sectors and found different results depending on the investigated sector. Somers and Nelson [15] argue that quality of the decisions in manufacturing that are related to technology, workforce, quality, production planning are important determinants of management perceptions of system value. Ettlie, Perotti, Joseph and Cotteleer [16] investigated the ERP adoption performance moderated by the type of industry, in particular manufacturing and services.

2.2 ERP Implementation-Related Strategies

In this section, we describe the variables that represent the strategies and activities that take place during the implementation process. These variables include process-system adaptation, support to implementation, vendor origin and temporal and scope implementation strategies.

Process-system adaptation. The simultaneous adaptation of technology and organization is a common topic in the literature on ERP implementation. Brehm, Heinzl and Markus [17] point out that companies that implement ERP systems have to decide whether to tailor the system to their organization or to adapt the organization so that it fits the architecture of the system. The former approach is known in literature as the "modeling choice" and refers to the process of adapting the software to the company's processes. The latter refers to the process of redesigning the company's processes in

[1] Only correlations at the 0.01 significance level are reported.

line with the architecture and processes of the software (a strategy commonly referred to in the literature as the "blueprinting" or "vanilla" approach [18]). Some authors [19] consider blueprinting to be a safe approach that minimizes cost and time overruns. Compared to modeling, blueprinting does not take into account the organizational needs and requirements and forces the company to adopt to the system. Blueprinting is a potentially risky approach because no system will fully be able to meet the needs of the entire organization and implementation my may result in misalignments. Nowadays, blueprinting is being chosen by vendors as a critical part of their offerings for small and medium enterprises (SME). Blueprinting allows SME to keep curb implementation costs of enterprise systems (ES) such as ERPs and keep them in balance with their limited resources.

The modeling option is often strongly discouraged by vendors and consultants. Tailoring the system may cause problems at different stages of the ES life cycle. Typically, tailoring the ERP system results in a prolonged implementation process, increased maintenance costs, and potential difficulties at the moment of upgrading [20], [21]. On the other hand, authors like Davenport [22] and Lorenzo and Diaz [23] propose ES implementation models in which organizations can reconcile their process needs with the system functionalities. These models prescribe an iterative process of reviewing organizational needs and system functionalities that result in a gap analysis output. This output facilitates the definition of a mix of initiatives including organizational changes and system tailoring aimed at bridging the gap. For example, Geneva Pharmaceutical [24] undertook its SAP implementation by using a modeling choice.

Process modeling. Process modeling is a technique that allows organizations to make the process knowledge explicit. Process modeling refers to the common task of reengineering methodologies, software development, and quality accreditations. The key objectives of a process modeling exercise are twofold; to generate an accurate representation of the existing processes and to correctly project new processes [25]. This is also seen as a preparatory phase for process analysis in a process change initiative. A vast body of literature exists describes the methodologies, techniques and tools used in process analysis and modeling (e.g. [26], [27], [28], [29]).

Implementation support. Once companies have taken the decision to implement an ERP system, they have the option to implement that system using the capabilities that are available in-house or, in their absences, they can obtain the skills externally by hiring consultants to facilitate the ERP implementation and to capture knowledge from the experts [30]. Contracting consultants and/or recruiting employees with previous experience are argued to have a positive impact on the post implementation performance of the ERP. Evidence drawn from case studies supports this claim [31].

Vendor origin. One of the first and most important decisions to be made at the outset of ERP implementation projects is the selection of the system vendor. Companies can either opt for a single system vendor for the entire organization or bring together a selection of system vendors with expertise in different functions or processes. The first option is known the "integral choice," whereas the latter is commonly defined as the "best-of-breed" or BoB choice [32].

The advantages of the integral choice reported in the literature include simplified interfaces, an increased leverage with the system vendors, and a single skill set required by information technology (IT) staff. Many companies recognize that the benefits of an ES come from full integration. Despite what the advocates say, the integral choice does have disadvantages as well. Given the fact that is it difficult, if not impossible, to design a system that will fit any possible organizational design or function, misfits are likely to occur between the system and some of the companies' core processes. For some companies this may result in unacceptable losses of competitiveness [33], [34]. The integral choice conveyed Ericsson Spain, for example, to delay the implementation of the B2B procurement functionality until SAP added this module [35]. Although Ericsson assessed other vendors the advantages of the integral choice strategy were the decisive factor in choosing that [36].

The advantages of the alternative BoB choice are that companies can obtain the best modules from specialized system vendors for their core processes and so achieve more flexibility in the business process redesign and facilitate easier supply chain integration. The interfacing process is facilitated by the development of object oriented interfaces and software known as middleware. The recent development of the service-oriented architecture (SOA) seems to have encouraged easier adoption of the BoB choice. A main disadvantage of the BoB choice is the complexity at the moment of upgrading the system, as the middleware has to be upgraded each time one of the modules is upgraded [37], [38], [39], [40]. The BoB choice was used by Nestle when they decided to implement SAP for their back-office processes and Manugistics for their supply chain processes [41].

Temporal and scope implementation strategies. Aside from the choice of one or several system vendors, companies need to decide whether to implement all modules simultaneously or progressively over time. The arguments for choosing one option or the other are centered on the urgency of implementation and the need for obtaining results versus complexity of implementation and risks involve in the process. Parr et al. [42] describe this strategic decision as the module implementation strategy. Markus et al. [43] relate this dimension to the scope of the implementation and describe it in terms of "big bang" approach and of "phased rollout" approach. A progressive implementation was chosen in a public sector case reported by Ni and Kawalek [44]. Volkoff [45], and Chan and Swatman [46] report other cases of progressive implementations. On the other hand, some organizations prefer a riskier "burn the ships" approach that creates commitment in the organization, as in the case of a simultaneous implementation described by Brown and Vessey [47].

2.3 ERP Performance Outcomes

Measuring the performance of ERP systems is a difficult task and scholars have taken different perspectives in their attempts of doing so. Performance could be measured taking a perceptual perspective whereby final users evaluate whether in their opinion the implemented system is useful or not. Measures of perceived usefulness, perceived ease of use, and user acceptance of information technology have been developed by Davis [48] to explain a system's success. Second, more traditional operational perspectives have been taken to measure the ERP's performance. Other authors have used constructs that measure cost reduction, better coordination, quicker response time, more on-time deliveries, better customer satisfaction, or shorter financial close cycles to operationalize ERP performance [49], [50], [51], [52]. Still others have taken a financial perspective considering the effect of the ERP system on the company's long-term financial performance by measuring the ROA, ROI, the return of sales, the cost of goods divided by sales, among others [53], [54]. Some authors [55] evaluate the impact of ERP systems by measuring the change in market value of firms after implementation. Finally, the level of ERP infusion has been the perspective followed by others such as Rajagopal [56], and Lorenzo and Kawalek [57].

3 Research Methodology

To explore the relation between the ex-ante business variables, ERP implementation strategies, and the perceived implementation contributions to business results we devised a survey instrument for collecting data. In this section, we first highlight our approach concerning the definition of the research population and the sample that we drew from. We then dedicate some words to the details of the development of the survey instrument.

3.1 Population, Sampling and Resulting Sample

Our target population comprised of Spanish and Latin American SAP users. Our specific focus on SAP users is motivated by the fact that SAP is widely considered the market leader in ERP solutions. Our focus on a single supplier was further driven by our concern that the wide variety of ERP solutions from a large selection of suppliers available would cause users of different systems to interpret the terminology used in the survey differently and thus introduce undesired noise in the data. By limiting our data collection efforts to the market leader SAP we assured sampling an objective cross-section of ERP users whilst minimizing the potential noise caused by factors beyond the specific focus of this study.

Early 2007, we invited 549 SAP users in Spain and Latin America to participate in our survey. Over the preceding months 105 companies responded positively to our invitation and returned the filled out questionnaire; providing a respectable gross response rate of 19%. We were forced to remove 14 companies from our sample due to unacceptable levels of missing data, reducing the net response rate to 16.6% (91 valid responses), still comparing favourably to the response rates reported by Spanish standards [58], [59]. Geographically one third of the respondents originated from companies based in Spain; 57% originated from South America (Argentina, 14%; Venezuela, 12%; Colombia, 11%; Chile, 9%; Peru, 5%). The remaining 12% were received from companies in Central and North America (Mexico, 11%, Guatemala 1%.)

In line with the SAP industry classification, 60% of the surveyed companies are from manufacturing and 40% from services (including financial institutions, utilities, insurance companies, distribution, logistics, department stores, media and professional services). In accordance with the SAP size classification, 10.5% of companies in the sample are classified as small (1 to 100 employees), 26.3% are medium (100 to 500 employees) and 63.2% are large (over 500 employees). The typical company in our sample is large, employing on average 3.336 local employees and reporting revenues of € 573 million per year.

Figure two provides an overview of the SAP modules that were implemented in the companies of our sample as well as the degree to which they were implemented (in %). The finance, purchasing, inventory, asset management and sales modules, for

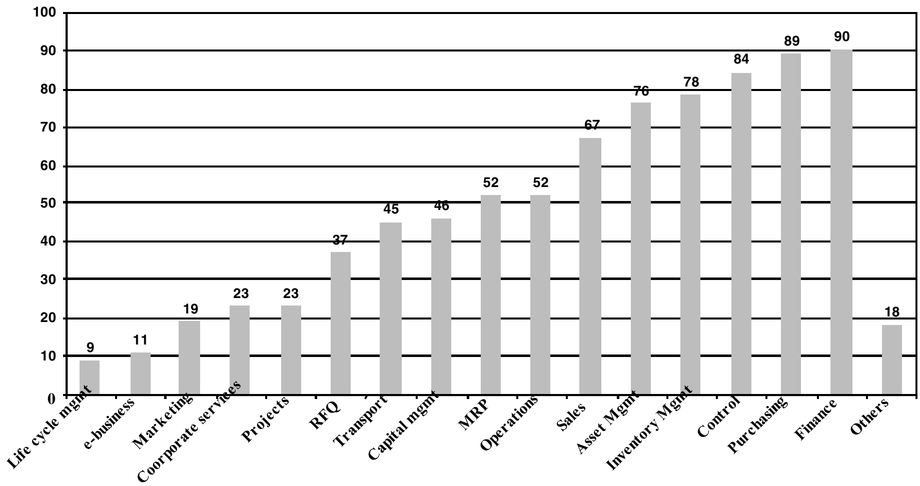


Fig. 2. Business areas and processes where SAP is used

example, were all implemented in over 75% of the companies. Conversely, modules for product life cycle management, electronic commerce, marketing, maintenance, projects and human resources were all implemented in less than 25% of the surveyed companies.

Half the companies used the classic version of SAP R/3 4.6. The vast majority (83%) of implementations had taken place in the operation environment between two and six years ago, 11% of the respondents had only recently implemented the ERP (less than two years of operation) and 6% were using the ERP for more than 10 years.

3.2 Respondents

Questionnaires were addressed to the individuals who, within reason, were expected to have the best (most relevant) overall picture of the firm; IT directors or general managers. A comprehensive understanding of the firm's entire operations was deemed important because ERP implementations do not tend to focus on one or few functional areas, but rather have an impact on the processes that are performed throughout the company. Because the survey was mostly submitted to medium and large sized companies, it would not be unusual to come across responses from lower level technicians who would have the knowledge and skills to provide reliable answers of particular parts of the ERP system, yet who lack the overview of the entire project and its objectives. IT directors or general managers, on the other hand, would be able to retrieve the required information from his or her subordinates if so needed.

3.3 Questionnaire Design, Pre-test and Administration

To collect data from the selected target population a web base survey instrument was designed, closely following the suggestions and experiences described in Dillman [60] Questions on business strategy, implementation results and impact of implementation on business results were anchored on 5-point Likert scales. The survey instrument was pre-tested in five firms representing the different industrial environments that were included in the population. As a result, changes were made to approximately 20% of the questions.

To enhance the response rate, a panel consisting of Ph.D. students and program assistants were instructed to contact non responding companies to motivate them to fill out the online survey instrument. All questionnaires have been systematically controlled for missing values and inconsistencies in an effort to ensure data quality and completeness.

4 Analysis of Results

Tables 1a, 1b and 1c show the variables in the conceptual model were operationalized, as well as the descriptive resulting from the survey data.

Table 1a. Operationalization of the ex-ante implementation variables and descriptive

Ex-ante implementation variables	Anchors	Ave.	Std.
Business Orientation - Low Cost	Not imp-V. imp	4.02	0.96
Business Orientation - Unique Product or Service	Not imp-V. imp	3.92	1.19
Business Orientation - Specific Market Segment	Not imp-V. imp	3.73	1.11
Top Management Support for IT Efforts	Low-High	3.81	0.97
Process Orientation	Low-High	3.45	0.67
Use of Consultants Prior to ERP Implementation	Low-High	3,94	1.15
Prior ERP implementation Experience	Low-High	2.07	1.23
Prior Process Modeling Experience	Low-High	2.49	1.31

Table 1b. Operationalization of the implementation variables and descriptive

ERP implementation strategies	Anchors	Ave.	Std.
Use of Consultants During the Implementation	Low-High	4,42	0.75
Use of Consultants After The Implementation	Low-High	3,48	1.2
Pre-implementation modeling of process	Low-High	3.46	1.18
Origin of Modules (Single / Multiple System Vendor)	Single-BoB	1.74	0.95
Temporal implementation strategy (Progressive/ All At Once)	Slow-Fast	3.48	1.47
Scope Implementation Strategy (Single Modules or The Entire Organization)	Single-All	4.11	0.77
Process-System Matching (Blueprinting: Adjust Process to ERP)	Low-High	3.86	1.04
Process-System Matching (Modelling: Adjust ERP to Process)	Low-High	3.03	1.1

4.1 Ex-ante Variables

Business orientation. Parity was observed in the approach that companies take concerning their competitive strategy. Two thirds of the companies consider a low operational cost strategy as important or very important. At the same time, 68.2% of the companies considered a strategy focussing on having unique products or services to be important or very important (note that a company could follow both strategies for different product lines.) Those companies that follow a dominating cost strategy

Table 1c. Operationalization of the ERP performance outcomes and descriptive

ERP performance outcomes	Anchors	Ave.	Std.
ERP Implementation On Time	Less- More	3.34	0.9
ERP Implementation Within Budget	Less- More	3.56	0.86
ERP implementation Within Scope	Less- More	3.16	0.72
Short Term Increase in Income	Not-A lot	1,94	1,02
Short Term Reduction of Inventory	Not-A lot	2,52	1,11
Short Term Reduction of Employees	Not-A lot	2,01	1,12
Short Term Reductions in Cycle Times	Not-A lot	2,95	1,06
Short Term Improvement on Customer Service	Not-A lot	2,69	1,14
Short Term Improvement in Supplier Relations	Not-A lot	2,82	1,08
Short Term Improvement of Capacity to Accommodate Changes	Not-A lot	3,11	1,15
Short Term Improvement in Decision Making Capacity	Not-A lot	3,21	1,11
Short Term Improvement in Coordination Between Areas	Not-A lot	3,35	1,03
Long Term Increase in Income	Not-A lot	2,33	1,03
Long Term Reduction of Inventory	Not-A lot	3,07	1,11
Long Term Reduction of Employees	Not-A lot	2,32	1,26
Long Term Reductions in Cycle Times	Not-A lot	3,39	0,90
Long Term Improvement on Customer Service	Not-A lot	3,16	1,08
Long Term Improvement in Supplier Relations	Not-A lot	3,41	0,99
Long Term Improvement of Capacity to Accommodate Changes	Not-A lot	3,74	1,00
Long Term Improvement in Decision Making Capacity	Not-A lot	3,78	0,90
Long Term Improvement in Coordination Between Areas	Not-A lot	3,79	0,84
Perception of Implementation Utility	Little-Much	3.93	0.6
Perception of User Satisfaction	Little-Much	3.55	0.6
Perception of Degree of System Adjustment to Company Processes	Little-Much	3.59	0.69

do not show a dominating implementation strategy; however, correlation was found between companies that follow a differentiation strategy and modelling implementation strategies.

*Previous Organizational Capabilities***.** Fifty five percent of surveyed companies had little, or very little, previous experience in implementing similar projects. This percentage is reduced in the case of service companies. Previous ERP implementation experiences and previous process modelling experiences are strongly correlated and load into a single factor (Previous experiences), with a Cronbach alpha of 0.783 [61], [62], [63]. Although this factor was found to have no effect on implementation results, it shows a negative correlation to blueprinting implementation strategies, suggesting that those with more extensive previous experiences avoid this strategy.

Coordination mechanisms. Sixty five percent of companies surveyed indicated that the involvement of top management and decision processes related to information technology is high or very high. Upper management participation was found to be correlated to a process orientation, and to modelling implementation strategies (but not to blueprinting implementations), suggesting that those implementation that require organizational change (as implicit in the modelling effort) do require top management involvement.

Organizational and Market context. The data analysis was controlled for possible effects of context variables such as size or industry sector. Size was evaluated by the number of employees and the revenue at the local facility (i.e., at the site of the implementation) and at the national level. Following SAP' standards respondents were classified into one of three sectors: Finance and public sectors, manufacturing and services. Three variables used to measure size, number of employees at the local level, at the national level, and revenue at the local level loaded into a factor with a Cronbach alpha of 0.780, that was used to control for size. No effect of size or industry was found on the observed results. We acknowledge that this may be due to sample bias, as the majority of the companies in the sample were large multinationals.

4.2 ERP Implementation-Related Strategies

Process-system orientation: Figure 3 shows that 69.3% of companies made major efforts to align their processes to the management system (blueprinting strategy). On the other hand, efforts to adapt the ERP to the business processes (modelling strategy) were reportedly smaller (right graph). As mention above, business differentiation strategies were found to be correlated to modelling-based implementations. While no relation was found between blueprint strategies and implementation results, there exist a significant correlation between modelling implementation strategies and implementation results, both in economic contributions and in completing the implementation within budget.

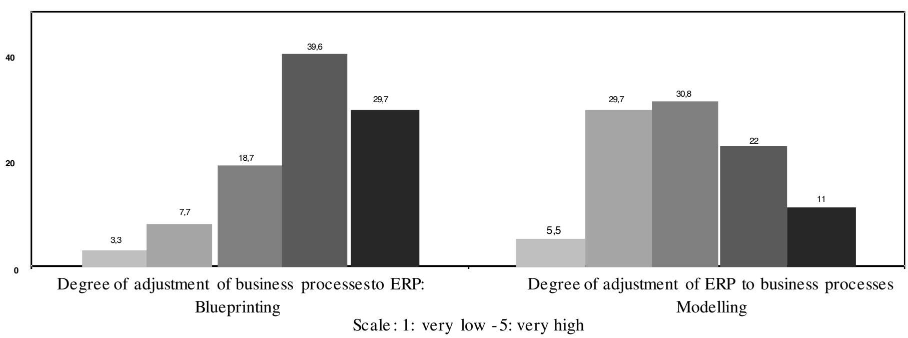


Fig. 3. Blueprinting and Modelling implementation strategies

Process modelling. Pre-implementation process mapping effort was high to very high in 50% of all cases. Process mapping was found to be correlated to long term economic and strategic implementation contributions to business results, both for blueprinting and modelling implementations, suggesting that implementation efforts should focus on this activity.

Implementation support. External consultants were widely used in all phases of the ERP implementation (pre, during and after), although with less intensity in the planning and in the post-implementation phases. No significant evidence of relationship between the use of external consultants and implementation success was found [64].

Vendor origin. A high percentage of companies surveyed (81.3%) opted for a single ERP supplier (little or no mix of applications from different providers). We acknowledge that the fact that the survey was aimed at users of SAP may influence this result. However, Best of Breed (modules from different vendors) implementations were positively correlated to Supply chain results, suggesting that inter-organizational applications may require functionalities beyond those provided by a typical ERP implementation (e.g., Manugistics or i2.)

Temporal and scope strategies. Fifty four percent of respondents followed a big-bang implementation strategy (shock, or rapid) and 80% a global scope implementation strategy (comprehensive implementation in the entire organization). Although these two strategies are correlated (suggesting that they tend to go together: fast and comprehensive implementation), no relation to implementation results was found.

4.3 ERP Performance Outcomes

The ERP implementation outcome was measured with three sets of variables: project results, customer satisfaction, and perceived contributions to business results.

Project results were measured in terms of time, budget and scope compliance. Forty percent of respondents exceeded the estimated implementation time, while 52% exceeding the estimated budget. Compliance in scope performed better, with only 16.5% below the expected scope. Scope compliance was the only project measurement that has a positive correlation with contributions to business results, suggesting that is more important to complete the implementation in scope than in time or budget.

The degree of satisfaction of the companies with the ERP implementation was measured by asking respondents how the implementation was perceived in terms of usefulness, user satisfaction, and degree of adjustment to the processes of the company. A large majority of respondents (81.3%) have a positive perception of the usefulness of the ERP, although user satisfaction and the adequacy of the ERP to business processes are markedly lower.

Perceived contributions to business results: Measure Development and Validation. Given the limited research into the perceived contribution of an ERP implementation, we were unable to identify any satisfying measures in the literature and hence developed a set of new measures With that aim in mind we presented a selection of items describing most common business objectives the sample population (Table 1). Respondents were then asked to indicate to which extent the implemented ERP system had contributed to achieve each of the different business objectives; (1) no contribution to (5) large contribution. Separate sets of items were presented for the short-term effects (up to three months after the implementation) and the medium term (between three and six months after implementation to allow for testing the effect over time. We tested the reliability of these measures in the form of internal consistency using Cronbach's alpha coefficient. Construct validity in the form of construct univocity was checked through principal component analysis (see Table 2) whereby all proposed construct complied with the eigenvalue greater-than-one criterion in which only those factors that account for variance greater than one should be included [65].

In Table 2 the construct *Economic Performance* (EP) refers to the extent to which the implementation of the ERP system has contributed to achieving the company's economic business objectives. ECP was operationalized by asking the respondents to state (both for short and medium term) whether the ERP implementation has contributed to (1) an increase in income, (2) a decrease in inventory and (3) a reduction of labour requirements.

The construct *Strategic Business Objectives* (STP) refers to the extent to which the implementation of the ERP system has contributed to achieving the company's strategic business objectives. STP was operationalized by asking the respondents (both for short and medium term) whether the ERP implementation has contributed to (1) reduce the product cycle times, (2) improve the company's capacity to accommodate changes in the market, and (3) improve the company's capacity to make better decisions.

The construct *Supply Chain Objectives* (SCP) refers to the extent to which the implementation of the ERP system has contributed to achieve the company's supply chain business objectives. SCP was operationalized by asking the respondents (both for short and medium term) whether the ERP implementation has contributed to (1) improve the company's customer service, (2) improve the company's supplier relations, and (3) improve the coordination between the company's different functional areas.

Table 2. Performance Construct validity

Construct	Full name / Explanation	Cronbach Alpha	# of items	1st eigenvalue	2nd eigenvalue	% of variance explained by 1st factor
EP	Economic Performance	0.828	6	3.263	1.068	54.4%
EP_ST	Short-Term Economic Performance (up to 3 months after implementation)	0.738	3	1.982	0.618	66.1%
EP_LT	Long-Term Economic Performance (3-12 months after implementation)	0.667	3	1.811	0.643	60.4%
STP	Strategic Performance	0.841	6	3.355	0.087	55.9%
STP_ST	Short-Term Strategic Performance (up to 3 months after implementation)	0.806	3	2.164	0.440	72.1%
STP_LT	Long-Term Strategic Performance (3-12 months after implementation)	0.769	3	2.055	0.506	68.5%
SCP	Supply Chain Performance	0.856	6	3.526	0.852	58.8%
SCP_ST	Short-Term Supply Chain Performance (up to 3 months after implementation)	0.853	3	2.323	0.460	77.4%
SCP_LT	Long-Term Supply Chain Performance (3-12 months after implementation)	0.714	3	1.923	0.619	64.1%

5 Discussion

Based on the analysis presented in this paper we induce some initial conclusions on business strategies, ERP implementation strategies and business performance. These are summarized in Table 3 and discussed below.

In our sample we observed a clear divide between pursuing a business strategy characterized by a dominant focus on low cost (price based) competition and companies pursuing a strategy characterized by product or market differentiation. Intuitively, the use of transactional modules suggests cost-base competition as the dominant strategy. However, while companies that follow cost based strategies could adopt either blueprinting or modelling implementation strategies, companies that follow a differentiation strategy tend to adopt process-based implementations (modelling). This observation reinforces the proposition by Lorenzo and Diaz [66] of adapting the ERP to the reengineered company business processes in the case of differentiation-based strategies. Our data shows furthermore that modelling strategies are characterized by a higher involvement of top management.

Table 3. Summary of main findings

Finding one: Companies that focus on differentiation tend to go for process based implementations (modeling); those that compete on cost don´t have a preferential implementation strategy
Finding two: The implementation strategies that better explain implementation results are those of previous process mapping and a modelling implementation strategy (which requires top management support). Other strategies, such as the use of consultants, blueprinting implementations, temporal and scope implementation strategies and previous experiences were found not to have a significant impact on the implementation results
Finding three: Exceptions to the previous were that Best of breed implementation may be required for SC results, and that companies with previous experiences tend not to go for blueprinting implementations
Finding four: Most widely perceived benefits of ERP implementations were related to strategic issues (e.g., capacity to accommodate changes, for decision making and for shorter cycle times) and to the supply chain performance (improved customer service, supplier relations and coordination between areas), rather than economics (e.g., inventory reduction, sales).
Finding five: Completing the scope of the implementation contributes to the implementation business results (in all dimensions: economic, strategic and supply chain), while completing it within time or budget does not.

Process modelling as a preparatory exercise before the ERP implementation was found to correlate significantly with better results in the implementation. Interestingly no evidence of better results was found for acquiring external expertise (i.e. consultants) in the implementation process. Similarly, neither prior experiences in process improvement and in ERP implementations nor the use of different temporal and scope strategies showed any positive relation with the implementation results.

The perceived contribution of the ERP implementation to business objectives can be categorized in three groups: economic objectives, strategic objectives and supply chain (or integration) objectives. The greatest perceived contributions were found to process integration, and the lowest to economic performance. In all cases, the perceived contribution increases a few months after the implementation. These results reinforce the acclaimed role of ERP systems as a facilitator of process integration, but also the awareness that the integration requires time to be assimilated by the organization [67], [68] and the realization of the anticipated benefits.

Despite that fact that many ERP implementation projects exceed both the estimated time frame and budget, the, results of the implementations from the perspective of user satisfaction and perceived usefulness are generally good.

In sum, our analysis allows us to induce general strategic rules for ERP implementations. Companies perceive better results from the ERP implementation when processes are mapped and potential solutions are modelled prior to implementation. Our data shows furthermore that the exercise of process modelling is positively related to business differentiation strategies and that top management support is crucial for a successful implementation (Figure 4). A key learning then to take home from our analysis is that *Process analysis matters.*

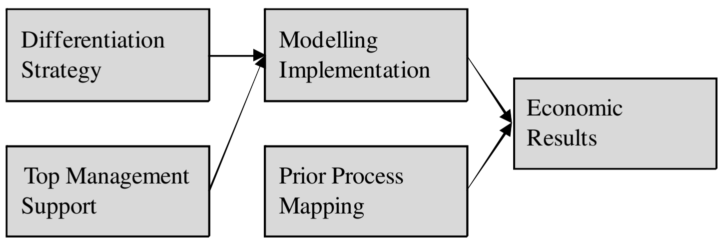


Fig. 4. Induced success implementation map

The relations suggested in Figure 4 were tested by running to two regression analysis. The first regression, of dependent variables "Differentiation Strategy" and "Top Management Support" with independent variable "Modelling Implementation" is significant at the 0.005 level and results in an adjusted R square of 0.124. The second regression, of dependent variables "Modelling Implementation" and "Prior Process Mapping" with independent variable "Economic Results" is significant at the 0.003 level and results in an adjusted R square of 0.130. The results, though significant, show modest values of R square. We attribute this to the high likelihood that other environmental factors are influencing the results, as well suggesting the need for future extension of the model through further research.

6 Limitations and Further Research

The research described in this paper is principally explorative. As a consequence threesome limitations to our findings have to be taken into consideration. The data that was collected for this paper, the instrument that was developed to do so and the results drawn from them should be considered as stepping-stones towards a more comprehensive approach of analysing the environmental aspects that influence the ERP implementation success. This paper provides relevant and valuable insights into the nature of the ex-ante business variables, internal process orientation, ERP implementation strategies, and the contribution of ERP implementation to the business results as suggested in Figure 4. It also makes a contribution to the literature by developing constructs to measure the (perceived) contributions of an ERP implementation to business results.

Future research should expand the research efforts initiated in this study and apply the findings to the refine the model presented in this paper into a model that can both clarify research issues on ERP implementation and offer valuable guidelines to practitioners.

We acknowledge the inherent bias caused by restricting the research sample to SAP implementations, and to the geographical area of Spain and Latin-America. Further research efforts should expand the geographical focus of the sample population to add strength to any findings that may be drawn for this emergent body of research.

References

1. Lorenzo, O., Díaz, A.: Process gap analysis and modelling in enterprise systems. International Journal of Simulation and Process Modelling 1(3/4) (2005)
2. De la Puerta Gonzalez-Quevedo, E.: El impacto de internet en las operaciones: del MRP a los marketplaces. Información Comercial Española 791, 33–54 (2001)
3. Davenport, T.: Mission Critical: Realizing the Promise of Enterprise Systems. Harvard Business School Press, Boston (2000)
4. Bhattacherjee, A.: Beginning SAP R/3 Implementation at Geneva Pharmaceuticals. Communications of the Association for Information Systems, 4 (2000)
5. Markus, M., Tanis, C.: The Enterprise Systems Experience - From Adoption to Success. In: Zmud, R. (ed.) Framing the Domains of IT Research: Glimpsing the Future Through the Past, Pinnaflex Educational Resources, Cincinnati (2000)
6. Parr, A., Shanks, G., Darke, P.: Identification of Necessary Factors for Successful Implementation of ER Systems. In: New Information Technologies in Organizational Processes - Field Studies and Theoretical Reflections on the Future of Work. Kluwer Academic Publishers, Dordrecht (1999)
7. Lorenzo, O., Díaz, A., Claes, B.: A Descriptive Analysis of the Antecedents of ERP Implementation. In: Proceedings of the EurOma Conference, Groningen (2008)
8. Porter, P.: Competitive Strategy: Techniques for Analysing Industries and Competitors. Free Press, New York (1980)
9. Park, J., Suh, H., Yang, H.: Perceived Absorptive Capacity of Individual Users in Performance of Enterprise Resource Planning usage: The Case for Korean Firms. Information and Management (44), 300–312 (2007)
10. Ko, D.G., Kirsch, L.J., King, W.R.: Antecedents of knowledge transfer from consultants to clients in enterprise system implementations'. MIS Quarterly 29(1), 59–85 (2005)
11. Parr, A., Shanks, G., Darke, P.: Identification of Necessary Factors for Successful Implementation of ER Systems. In: New Information Technologies in Organizational Processes - Field Studies and Theoretical Reflections on the Future of Work. Kluwer Academic Publishers, Dordrecht (1999)
12. Malone, T., Crowston, K.: The Interdisciplinary Study of Coordination. ACM Computing Surveys 26(1), 87–119 (1994)
13. Mabert, V.A., Soni, A., Venkataramanan, M.A.: The impact of organization size on enterprise resource planning (ERP) implementations in the US manufacturing sector. Omega 31, 235–246 (2003)
14. Everdingen, Y., van Hillegersberg, J., Waarts, E.: ERP adoptation by European midsize companies. Communications of the ACM 43(4) (2000)
15. Somers, T.M., Nelson, K.G.: The impact of strategy and integration mechanisms on enterprise system value: Empirical evidence from manufacturing firms. European Journal of Operational Research 146 (2003)

16. Ettlie, J., Perotti, V., Joseph, D., Cotteleer, M.: Strategic predictors of successful enterprise system deployment. International Journal of Operations and Production Management 25(9/10), 953–972 (2005)
17. Brehm, L., Heinzl, A., Markus, M.L.: Tailoring ERP Systems: a spectrum of choices and their implications. In: Proceeding of the 34th Annual Hawaii International Conference on Systems Science (2000)
18. Brehm, L., Heinzl, A., Markus, M. L.: Tailoring ERP Systems: a spectrum of choices and their implications. In: Proceeding of the 34th Annual Hawaii International Conference on Systems Science (2000)
19. Soh, C., Sia, S.K.: The Challenges of Implementing "Vanilla" Versions of Enterprise Systems. MIS Quaterly Executive 4, 373–384 (2005)
20. Brehm, L., Heinzl, A., Markus, M.L.: Tailoring ERP Systems: a spectrum of choices and their implications. In: Proceeding of the 34th Annual Hawaii International Conference on Systems Science (2000)
21. Light, B.: The Maintenance Implications of the Customization of ERP Software. Journal of Software Maintenance: Research and Practice 13, 415–429 (2001)
22. Davenport, T.: Mission Critical: Realizing the Promise of Enterprise Systems. Harvard Business School Press, Boston (2000)
23. Lorenzo, O., Díaz, A.: Process gap analysis and modelling in enterprise Systems. International Journal of Simulation and Process Modelling 1(3/4) (2005)
24. Bhattacherjee, A.: Beginning SAP R/3 Implementation at Geneva Pharmaceuticals. Communications of the Association for Information Systems 4 (2000)
25. Lin, F., Yang, M., Pai, Y.: A generic structure for business process modelling. Business Process Management Journal 8(1), 19–41 (2002)
26. Curran, T., Ladd, A. (eds.): SAP R/3 Business Blueprint. Understanding Enterprise Supply Chain Management. Prentice-Hall, USA (2000)
27. Succi, G., Predonzani, P., Vernazza, T.: Business Process Modelling with Objects, Costs, and Human Resources. In: Bustard, D., Kawalek, P., Norris, M. (eds.) Systems Modelling for Business Process Improvement. Artech House, USA (2000)
28. Harmon, P.: Business Process Change. Morgan Kaufmann, San Francisco (2003)
29. Kettinger, W., Teng, J., Guha, S.: Business Process Change: A Study of Methodologies, Techniques and Tools. MIS Quarterly (March 1997)
30. Somers, T.M., Nelson, K.G.: The impact of strategy and integration mechanisms on enterprise system value: Empirical evidence from manufacturing firms. European Journal of Operational Research 146 (2003)
31. Bhattacherjee, A.: Beginning SAP R/3 Implementation at Geneva Pharmaceuticals. Communications of the Association for Information Systems 4 (2000)
32. Light, B.: The Maintenance Implications of the Customization of ERP Software. Journal of Software Maintenance: Research and Practice 13, 415–429 (2001)
33. Light, B., Holland, C., Wills, K.: ERP and Best of Breed: A Comparative Analysis. Business Process Management Journal 7(3), 216–224 (2001)
34. Geishecker, L.: ERP Best-of-Breed. Strategic Finance, 63–66 (1999)
35. Lorenzo, O., Díaz, A.: Enterprise Systems as an enabler of fast-paced change: The case of B2B Procurement in Ericsson. In: ERP for Global Economies: Managerial Issues and Challenges. IGI Global (2008)
36. Computerworld (2002)
37. Davenport, T.: Mission Critical: Realizing the Promise of Enterprise Systems. Harvard Business School Press, Boston (2000)

38. Light, B., Holland, C., Wills, K.: ERP and Best of Breed: A Comparative Analysis. Business Process Management Journal 7(3), 216–224 (2001)
39. Markus, M., Tanis, C.: The Enterprise Systems Experience - From Adoption to Success. In: Zmud, R. (ed.) Framing the Domains of IT Research: Glimpsing the Future Through the Past. Pinnaflex Educational Resources, Cincinnati (2000)
40. Brown, C., Vessey, I.: NIBCO's Big Bang. Communications of the Association for Information Systems 5 (2001)
41. Worthen, B.: Nestle's ERP Odyssey. CIO Magazine (May 2002), `http://www.cio.com/archive/051502/nestle.html`
42. Parr, A., Shanks, G., Darke, P.: Identification of Necessary Factors for Successful Implementation of ER Systems. In: New Information Technologies in Organizational Processes - Field Studies and Theoretical Reflections on the Future of Work. Kluwer Academic Publishers, Dordrecht (1999)
43. Markus, M.L., Tanis, C., van Fenema, P.: Multisite ERP implementations. Communications of the ACM 43, 42–46 (2000)
44. Ni, Y., Kawalek, P.: When the Dust Settles: A Study of Users after Implementation of an ES in a Local Government Authority. In: Proceedings of The European Conference On E-Government (2001)
45. Volkoff, O.: Configuring an ERP System: Introducing Best Practices or Hampering Flexibility? Journal of Information Systems Education 14, 319–324 (2003)
46. Chan, C., Swatman, P.: Management and Business Issues for B2B e-Commerce Implementation. In: Proceedings of the 35th Annual Hawaii International Conference on System Sciences (2002)
47. Brown, C., Vessey, I.: NIBCO's Big Bang. Communications of the Association for Information Systems 5 (2001)
48. Davis, F.: Perceived Usefulness, Perceived Ease of Use, and User Acceptance of Information Technology
49. Poston, R., Grabski, S.: Financial impacts of enterprise resource planning implementations. International Journal of Accounting Information Systems 2, 271–294 (2001)
50. McAfee, A.: The impact of enterprise information technology adaptation on operational performance: an empirical investigation. Production and Operations Management 12(1) (2002)
51. Mabert, V.A., Soni, A., Venkataramanan, M.A.: The impact of organization size on enterprise resource planning (ERP) implementations in the US manufacturing sector. Omega 31, 235–246 (2003)
52. Gattiker, T., Goodhue, D.: What happens after ERP Implementation: Understanding the impact of interdependence and differentiation on plant level outcomes. MIS Quarterly 29(3) (2005)
53. Nicolaou, A.I., Bhattacharrya, S.: Organizational performance effects of ERP systems usage: The impact of post-implementation changes. International Journal of Accounting Information Systems 7, 18–35 (2006)
54. Hendricks, K., Singhal, V., Stratman, J.: The impact of enterprise systems on corporate performance: A study of ERP, SCM, and CRM system implementations Preview. Journal of Operations Management 25(1), 65–82 (2007)
55. Chavez, G., Lorenzo, O.: The impact of supply chain applications announcement on the market value of firms. Supply Chain Forum 7(2) (2006)
56. Rajagopal, P.: An innovation-diffusion view of implementation of ERP systems and development of a research model. Information & Management 40, 87–114 (2002)

57. Lorenzo, O., Kawalek, P.: A Model of Enterprise Systems Infusion. In: Proceedings of the European Operations Management Association (EurOma) Conference, INSEAD, France (2004)
58. Díaz, A., Solís, L., Claes, B.: Improving logistics and supply chain management practices in Spain; an analysis of current practices and future requirements. Journal of Logistics and Systems Management (Forthcoming)
59. Gallo, M., Villaseca, A.: Finance in Family Business. Family Business Review 9(4), 387–401 (1996)
60. Dillman, D.: Mail and Internet Surveys: The Tailored Design Method. Update with New Internet, Visual, and Mixed-Mode Guide. Wiley, Chichester (2007)
61. Cronbach, L.J.: Coefficient alpha and the internal structure of tests. Psychometrika 16, 297–334 (1951)
62. Nunally, J.C.: Psychometric Theory. McGraw-Hill, New York (1978)
63. Hull, C.H., Nie, N.: SPSS update 7-9. McGraw-Hill, New York (1981)
64. Lorenzo, O., Díaz, A., Claes, B.: Implantación de sistemas ERP: Factores de éxito e impacto en objetivos de negocio en empresas de España y Latinoamérica. Working paper IE, WP0833 (2008)
65. Norusis, M.J.: SPSS 13.0, Statistical Procedures Companion. Prentice Hall, New York (2005)
66. Lorenzo, O., Díaz, A.: Process gap analysis and modelling in enterprise Systems. International Journal of Simulation and Process Modelling 1(3/4) (2005)
67. Lorenzo, O., Kawalek, P.: A Model of Enterprise Systems Infusion. In: Proceedings of the European Operations Management Association (EurOma) Conference, INSEAD, France (2004)
68. Markus, M.L., Tanis, C., van Fenema, P.: Multisite ERP implementations. Communications of the ACM 43, 42–46 (2000)

A Technique for Annotating EA Information Models with Goals

Sabine Buckl, Florian Matthes, and Christian M. Schweda

Chair for Software Engineering of Business Information Systems (sebis),
Technische Universität München,
Boltzmannstr. 3, 85748 Garching, Germany
{sabine.buckl,matthes,schweda}@in.tum.de
http://wwwmatthes.in.tum.de

Abstract. Many of today's enterprises experience the need to establish and conduct management processes to ensure closely alignment between business and IT. Enterprise architecture (EA) management provides a model-based approach to understand and evolve the complex dependencies between the enterprise constituents, as e.g. business processes and business applications. In recent years the understanding of EA management in literature and in practice has converged, but up to this point no commonly accepted standard information model for EA management nor a standard set of goals verifying the overall objective of business/IT-alignment have been devised. Grounded in indications that such models and goals are highly enterprise-specific, this paper presents a method for flexible combining EA-relevant goals and EA information models to optimally support EA management in a using company.

Keywords: Enterprise architecture, Enterprise architecture management, Goal modeling, Modeling language, Goal question metric.

1 Introduction and Motivation

Many of todays large organization, first and foremost the globally acting enterprises, have to face the challenges of managing their business and IT structures as a whole. These structures commonly form highly interwoven systems, which mostly have developed for a long period and become increasingly complex trough mergers, business process re-organizations or technological shifts. Understanding the make-up of systems as well as of their interdependencies as *architecture*, it is sensible to allude to the aforementioned systems of constituents of the *enterprise architecture* (EA). Over the last years, practitioners and researchers have developed a variety of approaches for how to perform EA management and as one central task of this management process on how to document the EA. While in recent days some convergence in the understanding of EA as well as in the terms used to describe EAs can be diagnosed, the field is still far from having a common terminology or description technique. The latter especially applies in

J. Barjis (Ed.): EOMAS 2010, LNBIP 63, pp. 113–127, 2010.

the context of *EA management goals*, i.e. the goals that an organization seeks to attain by performing EA management. Some researchers as Aier et al. (cf. [1]) or Buckl et al. (cf. [2]) even challenge the hypothesis that a common description technique or a "standard" set of EA management goals can be devised. In contrast, they expect both techniques and goals to be highly organization-specific artifacts, accounting for the diversity of organizations as well as of their high level organizational goals and structure.

While one could expect that the aforementioned diversity is well-accounted by the different EA management approaches and modeling techniques, a survey on the current literature shows a slightly different image. Foreclosing the results of our discussion in Section 2 on the state-of-the-art in these areas, we summarize the situations as follows. EA description techniques are either

– developed without explicit reference to the concrete goals they are aiming at, most commonly resorting to abstract and general goals as "providing transparency", or
– tailored via distinct model concepts, as e.g. classes, to a narrow set of concrete goals without giving indications on how to adapt the concepts to an organization-specific utilization context.

Many EA literature does not specifically account for concrete EA management goals, but stays on the abstract and general level, as indicated above. If in contrast concrete goals are described, they are either given textually and without explicit reference to the EA model concepts, that they allude to, or directly incorporated into the corresponding modeling technique, e.g. via properties and KPIs. The second case nevertheless neglects the fact that the modeling technique might have to be adapted to be capable of satisfying the organization-specific information demands. From this, we derive the research gap that this paper aims to address. Summarized as a research question, the gap reads as follows:

> How can EA management goals be described in a manner that they can be incorporated into flexible EA modeling techniques?

The rest of the article is dedicated to answering the above question. Preparing our discussions, we revisit the state-of-the-art in EA modeling and EA management goal modeling in Section 2. Thereby, we seek to illustrate that the research gap is an actual one, but also provide a conceptual basis on which flexible EA modeling techniques are grounded. In Section 3, we present a technique that allows to model EA management goals as part of an EA information model. In this section, we further provide accompanying examples for the application of the technique. Concluding Section 4 summarizes critical points of the technique and provides an outlook on future research directions in this context.

2 State-of-the-Art in EA Modeling and Related Work

We approach the topic of modeling in the context of EAs from two different perspectives. At first, the *Goal Question Metric Approach* of Basili et al. (cf.

e.g. [3]) is introduced in Section 2.1 as a conceptual framework for reasoning on goals as well as the measurement of their achievement. Against that background, we secondly revisit the modeling techniques put forward by selected state-of-the-art EA management approaches in respect to the underlying information models as well as on the means for making EA management goals explicit in Section 2.2. Section 2.3 summarizes the current situation in literature and reifies the research question towards the description of the corresponding research gap.

2.1 The Goal Question Metric Approach

In [3] Basili et al. summarize the basic ideas of the Goal Question Metric (GQM) approach, developed some years earlier by Basili. Central to the approach is the understanding that goals, i.e. what is to be achieved, and metrics, measuring the achievement are closely related. Put in other words, goals are conceptual level entities, that are complemented by metrics on the quantitative level. The measurement of metrics is thereby understood as a mechanism to "aid in answering a variety of questions associated with the enactment of any [software] process". Later in [3] Basili et al. continue with "Measurement [...] helps to assess [the] progress [of a process], to take corrective action based on this assessment, and to evaluate the impact of such action". While the former quotations originate from the context of software development projects and processes, they also sensibly apply to many other processes targeting the creation (development) of a specific artifact. In this sense, we apply the basic notion of the GQM approach to EA management, which in line with e.g. van der Raadt and van Vliet (cf. [4]) can be understood as a management function aiming at the development of the EA.

In the GQM approach (cf. [3]), Basili et al. not only advocate for a strong linkage between goals and metrics but further emphasize that metrics, in order to be both sensible and applicable, must be "defined in a top-down fashion, [i.e.] focused based on goals and models". At this point, we should clarify the understanding of metrics in the GQM approach: a metric is quantitative information on a property of an object derived either via an *objective* measurement procedure or *subjectively*, e.g. by expert assessment. We will see below that according to this broad understanding metrics are widely used in EA management, although they are commonly not explicitly alluded to with the term "metric" but by the term "analysis". This term, while obviously correct, is usually used to mitigate some reservations in respect to the quantitative and objective nature of metrics. As shown later such reservations also exist in the context of EA management. To proactively dispel corresponding concerns, we provide the following working definition for the term "metric" based on the understanding of Basili et al.:

> A metric is at-least ordinally scaled information on a property of an object derived via an objective or a subject-dependent measurement procedure.

A side note has to be added to former definition concern the term "at-least ordinally scaled". By explicitly demanding this relatively lax *level of measurement* (for details on levels of measurement see Krantz et al. [5]), we allow e.g. that a property could be measured by a metric that evaluates to "high", "medium", or "low".

Coming back to the GQM approach of Basili et al. (cf. [3]), a last but highly important concept remains to be introduced: the *question*. Where the goal resides on the conceptual level stating what is to be achieved and the metric provides quantitative information on object properties, the questions comprise the "operational level" mediating between the two aforementioned levels. Put in words of Basili, questions "characterize the object of measurement" but do neither give measurement prescriptions nor indicate an intended achievement. In this sense, the questions establish the link between the goals and the objects that are affected by this goal, i.e. concretize the goal dependencies of the developed artifact. The questions further provide an abstraction from the concrete metrics used to quantify the achievement of goals.

2.2 EA Information and Goal Modeling

The Open Group Architecture Framework (TOGAF) [6] provides since its most recent version 9.0 the "Enterprise Content Metamodel" that describes the core classes, properties and relationships that make up an EA model. The content metamodel thereby puts strong emphasis on holistically covering structural aspects of the EA and is further designed as largely monolithic, i.e. should be used as a whole. A possible exception to latter design principle are the "extensions" that are described as part of TOGAF. These extensions define content metamodel fragments that can be added but do not have to be used. Two of these extensions, the so called "governance extension" and "motivation extension", are concerned with the aspect of goal modeling. Put in more detail, the extension introduce the concepts `Goal`, `Objective` and `Measure`, which are related to each other. These concepts are conversely not equipped with properties other than `name` such that only a statement as "the goal of *increased business continuity* effects (over the measure *availability*) the business service *settle credit card payments*". More detailed modeling of goals is not supported and the goals, more precisely their corresponding measures, cannot be linked to properties of the affected model concept, as `business service`.

The research group of Winter at the university of St. Gallen defines the "core business metamodel" (CBM) as basic model for EA modeling (cf. [7,8]). The CBM defines classes and relationships useful for describing the structure of an EA. The CBM is further designed as one comprehensive model, although recent publications concede (cf. [1,9]) that a demand to adapt the model to the specific requirements of a using organization exists. When it comes to the operationalization of goals into the model, three concepts `Strategic goal`, `Success factor`, and `Performance indicator` as well as the complementing relationships are introduced. The performance indicators can further be related to any type of reference object in the EA, but do conversely not provide means to express that one of the reference object's properties relates to the indicator. This may be ascribed to the fact that the CBM does not define properties, but this limit also applies to models that are derived from the CBM via adding properties, which is according to Kurpjuweit [8] a well-established technique for adapting the CBM. Summarizing, the model allows to relate goals to the underlying success factors as well as to operationalizing performance indicators. In contrast, the goals cannot be linked to the architectural properties in the model.

Niemann gives in [10] information models for describing different architectural layers in the EA, namely the "business", the "application", and the "system layer". On all these layers, core classes and relationships for describing the corresponding architectural concepts are introduced. Properties further describing the classes or the underlying architectural concepts, respectively, are only supplied for a few classes in the information models. Goals, more precisely EA management goals, are not part of the information model, while "business goals" can be described as part of the business layer. Complementing the information model, Niemann provides a plethora of analysis techniques that can be used to measure the attainment of different EA management goals. Most of the analysis techniques are thereby described textually with possible exemplification along graphical models of exemplary EAs. For a few analysis techniques, Niemann provides mathematical equations for deriving performance indicators from properties of the overall EA or the application landscape, respectively. For the topic of cost calculation, Niemann gives a basic economic equation summing up the different types of annual costs associated with an application system and a yearly (linear) deduction of investment costs. These costs are further mirrored as properties in the corresponding information model. When it conversely comes to other performance indicators, Niemann does not supply a link between the equations and the information model.

The architecture method "Quasar Enterprise" described by Engels et al. in [11] gives different information models describing parts of the overall EA. These information models consist of classes and corresponding relationships, but do not provide information on architectural properties that might be of relevance. `Business Goals` are modeled in Quasar Enterprise as part of the the strategy modeling and are interrelated to `Business Services` as part of the logical architecture of an application landscape. Further details on how to measure, to which degree a goal is attained, are neither given in the architecture method nor in the information models. Conversely, different techniques for analyzing the application landscape in respect to general quality measures are briefly sketched in the method. As an example, the quality measure "purity of domains" can be seen. This measure counts to which extent the current `Business Applications` supply `Business Services` for more than one future, i.e. target, `Business Domain`. While a relationship between the information model classes and the corresponding measurement rule is established textually, the information model does not reflect such calculations.

Gringel and Postina, as members of the research group of Appelrath at the OFFIS in Oldenburg, describe in [12] a reification of the architecture quality measures put forward in Quasar Enterprise. Put in more detail, they provide an information model adapted from the model of the Quasar Enterprise method complemented with a set of equations on how to derive values for corresponding quality measures, as e.g. "purity of domains". The correspondence between the the equations on the one hand and the information model concepts on the other hand is established via names, i.e. properties and concepts are named equally in both representations. The work of Gringel and Postina integrates well into other research results of the group of Appelrath as presented by Addicks and Steffens in [13] as well as by Addicks and Gringel in [14]. In the later publication, different "key figures", i.e. indicators, and their defining equations are presented. The equations nevertheless are not complemented with corresponding information models, such a potential user of the indicators might have difficulties to derive the architectural model, that the indicators are built upon.

Lankhorst et al. describe in [15] both a modeling language for EAs and complementing analysis techniques. The information model underlying the modeling language defines the classes and relationships of the architectural elements, but do not specify properties of the corresponding classes. In the analysis techniques, architectural models describing the structure of the EA are augmented with quantitative information, e.g. on "service execution times" or "interarrival times". These augmentations nevertheless take place on instance level, i.e. a concrete service is supplied with such information, while the class `Service` in the information model does not specify corresponding properties. Complementing the augmentations with quantitative information, expressions for deriving certain property values are supplied in mathematical equations.

In [16] Lankes and Schweda describe two "information model patterns" targeting the goal of *business continuity* by analyzing the property of availability. The information model patterns pick up the notion of the EA management pattern as introduced by Buckl et al. in [2] as building blocks for a flexible EA description technique. The complementing analysis proposed by Lankes and Schweda explores how failures of single `Business Applications` propagate through the EA ultimately rendering the execution of one or more `Business Processes` impossible. The information model contains the basic classes and relationships for describing the EA structure on which the analysis is performed. Complementing, the models are augmented with properties, as e.g. `failureProbability`, establishing the link to the corresponding goal. These properties are *derived* properties, for which the information model further supplies derivation rules formulated in the Object Constraint Language (OCL) [17] as well as in mathematical expressions. For the specific goal of business continuity and the selected architectural description language, the information model patterns of Lankes and Schweda [16] achieve a strong linkage, although the relationship between the goal and the underlying performance indicators are only described textually.

Johnson and Ekstedt provide in [18] a collection of information models of which each reflects a specific *viewpoint* on the overall EA. These information models introduce classes and relationships for describing a specific part of the EA, but do not supply properties for further specifying the corresponding instances. A dedicated "goal viewpoint" introduces the concept of the `Goal`, which can participate in a goal-hierarchy and can be linked to `Initiatives` for pursuing the goal as well as to `Problems` hindering the goal's achievement. Further relationships from the goal class are not provided, although the work introduces another modeling language, namely the *influence diagrams* specifically dedicated to goal modeling. An influence diagram is used to relate the central property of a goal to the architectural properties, which define this central property. Exemplifying this, we revisit the influence diagram defining what "performance" is meant to be composed of. The diagram makes explicit that "response time", "throughput" and "scalability" are definitorial for "performance", which conversely is central to any goal targeting the achievement of specific performance characteristics.

2.3 Summarizing the Research Gap

Reflecting the plurality of models for EAs as well as EA-relevant goals against the prefabrics of the GQM approach, we can elicit two common "patterns", how goals are incorporated into or linked to an EA model. The first pattern (GOAL-TO-ANY-OBJECT) typically contains a variant of the information model fragment shown in Figure 1. Such fragment can be found in the approaches of TOGAF [6], of Winter et al. [7], of Niemann [10], of Engels et al. [11], and of Johnson and Ekstedt [18]. In the understanding of the GQM approach, the pattern operationalizes a goal into the corresponding questions, more precisely into the characterization for the "objects of measurement" that the questions provide.

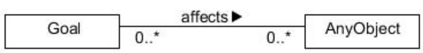


Fig. 1. Pattern GOAL-TO-ANY-OBJECT

The second pattern (METRICS-TO-INDICATORS) describes the fact that architectural properties are interrelated to indicators, more precisely used to define these indicators. This pattern can be found in the EA management approaches of Niemann [10], of Gringel and Postina [12], of Lankes and Schweda [16], and of Johnson and Ekstedt [18]. Against the conceptual framework of the GQM approach, this pattern is slightly more complicated to understand. Many of the architectural properties also are metrics, i.e. are per se at-least ordinally scaled. The properties that do not directly support this level of measurement, e.g. the nominal property used to indicate the *standard vs. custom software* nature of a business application, are in the context of the pattern METRICS-TO-INDICATORS supplied with additional measurement assumptions that allow to interpret them on a higher level of measurement. In the standardization example, such assumption would read as "standard is better than custom software", although in reality much more detailed measurement assumptions are employed. Put in other words, the pattern aggregates different measurable architectural properties (metrics) to a more coarse grained "way of assessment", i.e. a question in the sense of the

GQM approach. Different implementations for this pattern and thereby required relationships between architectural properties are given by the approaches:

Mathematical expressions are used by Lankhorst et al. [15], by Niemann [10], and by Gringel and Postina [12]. These expressions are highly expressive, but miss a linkage to the information model. Further, they may be formulated on an abstract level only using a non-further specified function as well as on a concrete and executable level.

OCL expressions are used by Lankes and Schweda [16]. OCL expressions are slightly less expressive than their mathematical counterparts but provide a strong linkage to the information model. Regarding the level of abstractness, only concrete, i.e. executable, OCL expressions can sensibly be formulated.

Influence diagrams are used by Johnson and Ekstedt [18]. Influence diagrams allow a concise and abstract description of relationships between architectural properties and relevant indicators, but are per se of limited expressiveness[1]. Additionally, influence diagrams do not provide means to link to the information model.

While each of the aforementioned patterns captures a part of the GQM trifecta, both patterns are on their own not able to implement the GQM approach in the context of EA modeling. With this more elaborate understanding of the context at hand, we can concretize the research question from Section 1 to a research gap as follows:

> EA management aims to develop and evolve the EA in the direction of EA-relevant goals. To support such development, an EA model should link each goal to the thereby affected objects, should operationalize goals into more concrete EA questions, and should provide measurable metrics for answering the EA questions in a quantitative way. How can we describe EA-relevant goals, questions and metrics in a way to achieve all of the former, while being flexible in respect to both the underlying EA information model and the concrete measurement as well as aggregation prescriptions?

Subsequent Section 3 is dedicated to explaining a technique for annotating EA information models with informations on EA management goals.

3 Introducing a Technique for Annotating EA Information Models with Goals

Before we introduce our technique for goal modeling in the context of EA modeling, we have to provide some clarifications and definitions that should help us to avoid ambiguities in the remainder of the section. In line with the understanding

[1] In [19] Johnson et al. describe how influence diagrams can be enriched with a more expressive semantics.

of the term "model" as put forward by Stachowiak in [20], we define EA modeling as the activity of creating "purposeful abstractions of the EA" with respect to their "intended usage context" and "using stakeholders". From this, we can emphasize on two central challenges of EA modeling, namely:

– abstracting the **right part** of the overall EA
– to support the **intended usage context** that corresponds to pursuing **intended EA-relevant goals**.

These challenges are closely related, but nevertheless yield two different perspectives from which EA modeling can be approached. Firstly, each EA model commits to a specific EA *concern*[2], i.e. area-of-interest in the overall architecture of the enterprise. Secondly, each EA model commits to at least one EA-relevant goal, i.e. provides information necessary for (measuring) the achievement of the goal. While proceeding towards our technique for goal modeling, this dichotomous nature of each EA model must kept in mind.

Taking a concern perspective, each EA model depicts a part of the architectural reality of an enterprise. Any EA model is expressed[3] using a distinct description language, the so-called *modeling language*. The necessity to have a modeling language complementing a model at first seems quite obvious, but finds further support when reasoning on *identity*. Van Leeuwen discussed in [22] that identifying a real world individual is only possible, if one associates a type to the individual and if the type supplies a conceptualization of identity. What might sound like an ontological sophistry is well exemplified by Guizzardi in [23] as follows: To know what the real world individual the term "Mick Jagger" refers to, we have to know the type of individual that we can draw our conception of identity from. So the identification of "Mick Jagger" becomes possible by knowing that we look for a `Man` "Mick Jagger". Guizzardi calls types that supply a conception of identity "sortals". Linking back to the concern perspective on an EA model, we can sensibly assume that each individual in the model is instance of a sortal, as e.g. `Business Application` or `Business Process`. These sortals, properties thereof, and the relationships between the sortals form the EA information model, which we – up to this point – have rather intuitively understood as meta-model for an EA model. Figure 2 exemplifies the concern perspective in EA information modeling, depicting the sortals `Business Application` and `Business Process`. To indicate the sortal nature of the corresponding classes, we utilize the UML profile stereotypes put forward by Guizzardi in [23].

From the usage context or goal perspective, each EA model reifies questions related to goals and provides corresponding quantitative answers via metric values. This calls for distinct characteristics in the models underlying modeling language. Put more precisely, a metric can be identified with a property in the

[2] The term *concern* is used here in line with its definition in the ISO Standard 42010 (cf. [21]).

[3] One might argue that this fact only holds for *explicit* models, i.e. models accessible to more than one person. For the purpose of this work, we do not regard this a relevant confinement.

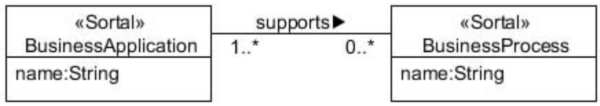


Fig. 2. Exemplary EA information model from concern perspective

EA model's corresponding information model. Picking up the working definition of metric from Section 2.1, we can further promote the constraint that a property reflecting a metric must be of a datatype supplying at least an ordinal scale. This conceptual modeling for metrics aligns well with the examples for metrics as given e.g. by Lankes and Schweda [16]. When it comes to the representation of a question in the EA model or the EA information model, respectively, the situation becomes a bit more intricate. The twofold nature of the question as

1. designator for objects of measurement, i.e. of objects that are affected by a certain goal, and as
2. aggregator for metrics that measure the achievement of the corresponding goal

aggravates a conceptualization of a question in the EA information model. To devise a suitable modeling construct, we employ a "linguistic trick" and substantiate the question into a conceptual type. For exemplifying this trick, we take the following question that results from the operationalization of a business process optimization goal:

> **(Q)** What is the performance (latency, throughput) of a business process?

In order to answer this question in an EA information model, the sortal `Business Process` has to be "performance-measurable", i.e. must supply properties reflecting the metrics "latency" and "throughput" which are used to define "performance". The, admittedly artificial, nomination of "performance-measurable" sheds an interesting light on a possible conceptualization of a question: a question literally adds an *attribute* to a corresponding *noun*, i.e. a sortal. In line with Guizzardi (cf. [23]) such attribution defines a specialized ontological type, the so-called *mixin*[4]. Mixins are *dispersive* types, i.e. specify sets of related (and co-located) properties, of which conversely none provides an identity to the thereby described type. Using a mixin type, we can mirror a question as a concept in the EA information model as shown in Figure 3. In the figure we again utilize the UML profile stereotypes put forward by Guizzardi in [23], here to indicate that a class represents a mixin type. We further employ a slightly adapted form of the notation of the *attribute dependency relationship* described by Buckl et al. in [24].

[4] In programming languages mixins are sometimes referred to as aspects. For reasons of clarity we abstain from overloading this term.

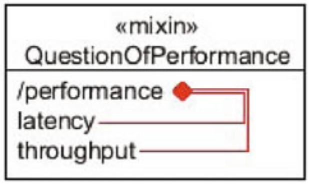


Fig. 3. Exemplary EA information model from goal perspective

Using the notation of the *attribute dependency relationship*, we display that the property "performance" is defined by the metrics "latency" and "throughput", while simultaneously abstaining from needlessly giving details on how to actually derive a performance measurement. This resembles the *definitorial relationship* in the influence diagrams of Johnson and Ekstedt (cf. [18]), but in contrast to the former allows a seamless integration into modeling technique used for EA information modeling. The properties of "performance", "latency" and "throughput" are further not assigned data types, as different measurement procedures for the two metrics will lead to different data types. In aggregating the metrics, more precisely their values, into a single quantitative answer to the question of performance, i.e. to a value for the derived "performance" property, also different options exist. The absence of a data type in the performance mixin accounts for this fact.

In order to consistently model the aforementioned question **(Q)**, we have to apply the mixin onto the sortal `Business Process` and have to supply data types for the corresponding metrics, namely for "latency" and "throughput". At this point, an integrated EA information model is created. Using this information model, an enterprise can answer the question of performance in a comparative manner, i.e. can determine for any two business processes A and B, if business process A is more, equally or less performant than business process B, or if no statement can be given. The integrated EA information model clearly indicates that such comparisons are possible, based on component-wise comparison of the two defining metrics "latency" and "throughput", which – according to the definition of the term metric from Section 2.1 – are each comparable. Figure 4 displays the metrics' values for exemplary business processes A to E, whereas complementing Table 5 displays the results of the corresponding performance comparisons. The symbol $\neq$ therein denotes that no comparison is possible.

Having integrated the `Question of Performance` mixin into the sortal `Business Process`, a goal- and concern-specific EA information model is developed. Aforementioned elementary comparison can further be used to provide decision support in evolving an EA, more precisely the business process support provided by the company's business applications. Nevertheless, this is not necessary the end of a development process for an enterprise-specific information model. A using enterprise may go beyond the basic characterization and select dedicated datatypes for the metrics "latency" and "throughput", for example "milliseconds" and "items per hour", respectively. Based on this information,

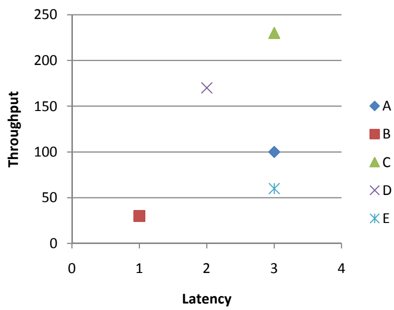


Fig. 4. Latency-throughput diagram

from \ to	A	B	C	D	E
A	=	≠	<	<	>
B	≠	=	≠	≠	≠
C	>	≠	=	≠	>
D	>	≠	≠	=	>
E	<	≠	<	<	=

Fig. 5. Comparison table

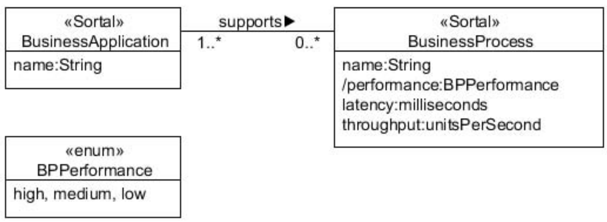


Fig. 6. Integrated information model from a goal and concern perspective

the definitorial relationships between these metrics and the property "performance" can be reified to a concrete computation prescription. Figure 6 shows the augmented EA information model resulting from the next development step.

The information model defines a datatype `BPPerformance` distinguishing three categories of business process performance. For determining, which value for the performance property applies, the following derivation rule reifying the *attribute dependency relationship* is supplied:

$$\text{performance} := \begin{cases} \text{low} & if \ \text{latency} > 10 \wedge \text{throughput} < 100 \\ \text{high} & if \ \text{latency} < 1 \vee \text{throughput} > 1000 \\ \text{medium} & else \end{cases} .$$

4 Critical Reflection and Outlook

In this paper we elicited how EA-relevant goals can be operationalized into questions, which are conversely represented as mixin types for annotating EA information models. With the mixins defining architectural properties, we could further show how concrete metrics are linked to the questions and incorporated into an EA information model. Thereby, a consistent modeling of EA-relevant goals, operationalizing questions and quantifying metrics could be achieved. An example building on the prefabrics of Johnson and Ekstedt [18] as well as of

Buckl et al. [2] provided first insights into the applicability of the presented technique.

Aside the example, applicability of the technique has not been subjected to further investigation. Especially, an analysis of the usefulness of the technique in a practical setting remains to be undertaken. Such analysis would be especially interesting, as the ability to "mix" questions into arbitrary EA information model sortals has the potential to lead to a plethora of combinations, of which not all might be sensible. Exemplifying this, one could think of the application of the aforementioned `Question of performance` mixin to the sortal of the `Organizational Unit`. While on a fairly abstract level, it might be sensible to assess the performance of an organizational structure, the metrics of `latency` and `throughput` may not be the most appropriate to do so. In the light of such unusual modeling that can result from applying the technique, it remains to be analyzed in a practical setting, if EA information models of that kind actually are created. Put more precisely, a practice cases have to show, if potential users of this modeling technique, can more easily create sensible models or if that kind of abstraction increases the danger to create "absurd", i.e. non-sensible, models.

In the paper also only a single EA information model and a single type of question have been analyzed. While this was sufficient to show, how the method could be applied, a broader analysis in respect to other information models, goals and corresponding questions is yet to be performed. Refraining the modeling of questions by Johnson and Ekstedt in [18], we are confident that the proposed technique can be applied to a broad variety of EA-relevant goals and their operationalizing questions, respectively. When it comes to EA information models, the pattern based method introduced by Buckl et al. in [2] and pertaining to a pattern language as described in [25] makes us confident that the technique devised in this paper can widely be applied in the context of EA modeling.

The pattern language further indicates a direction, into which mixin-based question modeling could evolve: future research could develop a collection or language of "question patterns". Such pattern would describe typical questions as repeatedly used in addressing EA-relevant goals together with best-practice quantifications of the questions, i.e. with practice-proven metrics for answering the questions. This further links to a tool for utilizing such collection of questions in designing an enterprise-specific EA information model. This tool should support the fragment- and mixin-based development and evolution of EA information models for practical usage contexts. With the help of such tool, the aforementioned danger of creating non-sensible information models could be analyzed in experiments or case studies.

References

1. Aier, S., Kurpjuweit, S., Riege, C., Saat, J.: Stakeholderorientierte dokumentation und analyse der unternehmensarchitektur. In: Hegering, H.G., Lehmann, A., Ohlbach, H.J., Scheideler, C. (eds.) GI Jahrestagung (2), Bonn, Germany. LNI, vol. 134, pp. 559–565. Gesellschaft für Informatik (2008)

2. Buckl, S., Ernst, A.M., Lankes, J., Schneider, K., Schweda, C.M.: A pattern based approach for constructing enterprise architecture management information models. In: Wirtschaftsinformatik 2007, pp. 145–162. Universitätsverlag Karlsruhe, Karlsruhe (2007)
3. Basili, V.R., Caldiera, G., Rombach, H.D.: The Goal Questin Metric Approach. Wiley, Chichester (1994)
4. van der Raadt, B., van Vliet, H.: Designing the enterprise architecture function. In: Becker, S., Plasil, F., Reussner, R. (eds.) QoSA 2008. LNCS, vol. 5281, pp. 103–118. Springer, Heidelberg (2008)
5. Krantz, D.H., Luce, D.R., Suppes, P., Tversky, A.: Foundations of Measurement. Additive and Polynomial Representation, vol. 1. Additive and Polynomial Representation. Academic Press, New York (1971)
6. The Open Group: TOGAF "Enterprise Edition" Version 9 (2009), http://www.togaf.org (cited 2010-02-25)
7. Österle, H., Winter, R., Hoening, F., Kurpjuweit, S., Osl, P.: Der St. Galler Ansatz des Business Engineering: Das Core Business Metamodel. Wisu – Das Wirtschaftsstudium 2(36), 191–194 (2007)
8. Kurpjuweit, S.: Stakeholder-orientierte Modellierung und Analyse der Unternehmensarchitektur. PhD thesis, Universität St.Gallen (2009)
9. Aier, S., Kurpjuweit, S., Schmitz, O., Schulz, J., Thomas, A., Winter, R.: An engineering approach to enterprise architecture design and its application at a financial service provider. In: Modellierung Betrieblicher Informationssysteme (MobIS 2008) – Modellierung zwischen SOA und Compliance Management, Saarbrücken, November 27-28, pp. 115–130 (2008)
10. Niemann, K.D.: From Enterprise Architecture to IT Governance – Elements of Effective IT Management. Vieweg+Teubner, Wiesbaden (2006)
11. Englebert, V., Heymans, P.: Towards more extensible metacase tools. In: Krogstie, J., Opdahl, A.L., Sindre, G. (eds.) CAiSE 2007 and WES 2007. LNCS, vol. 4495, pp. 454–468. Springer, Heidelberg (2007)
12. Gringel, P., Postina, M.: I-pattern for gap analysis. In: 2nd European Workshop on Patterns for Enterprise Architecture Management (PEAM 2010), Paderborn, Germany (2010)
13. Addicks, J.S., Steffens, U.: Supporting landscape dependent evaluation of enterprise applications. In: Bichler, M., Hess, T., Krcmar, H., Lechner, U., Matthes, F., Picot, A., Speitkamp, B., Wolf, P. (eds.) Multikonferenz Wirtschaftsinformatik, pp. 1815–1825. GITO-Verlag, Berlin (2008)
14. Addicks, J.S., Gringel, P.: Application landscape metrics: Overview, classification, and practical usage. In: 3rd International Workshop on Enterprise Modelling and Information Systems Architectures, Ulm, Germany, pp. 55–68 (2009)
15. Lankhorst, M.: Enterprise Architecture at Work: Modelling. In: Communication and Analysis. Springer, Heidelberg (2005)
16. Lankes, J., Schweda, C.M.: Using metrics to evaluate failure propagation and failure impacts in application landscapes. In: Multikonferenz Wirtschaftsinformatik. GITO-Verlag, Berlin (2008)
17. OMG: Object constraint language (ocl) available specification, version 2.2 (formal/2010-02-01) (2010)
18. Johnson, P., Ekstedt, M.: Enterprise Architecture – Models and Analyses for Information Systems Decision Making. In: Studentlitteratur, Pozkal, Poland (2007)

19. Johnson, P., Johansson, E., Sommestad, T., Ullberg, J.: A tool for enterprise architecture analysis. In: 11th IEEE International Enterprise Distributed Object Computing Conference (EDOC 2007), Annapolis, Maryland, USA, October 15-19, pp. 142–156. IEEE Computer Society, Los Alamitos (2007)
20. Stachowiak, H.: Allgemeine Modelltheorie. Springer, Wien (1973)
21. International Organization for Standardization: Iso/iec 42010:2007 systems and software engineering – recommended practice for architectural description of software-intensive systems (2007)
22. van Leeuwen, J.: Individuals and sortal concepts: an essay in logical description metaphysics. PhD thesis, University of Amsterdam (1991)
23. Guizzardi, G.: Ontological foundations for structural conceptual models. PhD thesis, CTIT, Centre for Telematics and Information Technology, Enschede, The Netherlands (2005)
24. Buckl, S., Ernst, A.M., Schweda, C.M.: An extension to the essential meta-object facility (emof) for specifying and indicating dependencies between properties. Technical report, Technische Universität München (2008)
25. Chair for Informatics 19 (sebis), Technische Universität München: Eam pattern catalog wiki (2010), http://eampc-wiki.systemcartography.info (cited 2010-02-25)

Applying DEMO in Operations Research: Lot Sizing and Scheduling

Roland W. Ettema

[1] Delft University of Technology, Netherlands
[2] Logica Consulting, P.O. Box 159, 1180 AD Amstelveen, Netherlands
Roland.Ettema@gmail.com

Abstract. As an interdisciplinary field, enterprise engineering can benefit from a wide spectrum of methods, including optimization methods used in Operational Research (OR). However, current optimization approaches (e.g. linear programming) in operations research take a narrow, mathematical view on the problem by simplifying the problem and often making assumptions. As a result, the obtained solutions may significantly differ from what could be obtained by applying a comprehensive enterprise engineering approach. To reduce the discrepancy between solutions obtained from an optimization approach and an enterprise engineering approach, and to obtain a more accurate solution of the problem, in this paper we propose to combine DEMO (an enterprise engineering approach) and Linear Programming (an OR approach). This combination not only helps to obtain accurate results but also to capture crucial views. We hypothesize that these approaches, in a combined manner, capture both the structure and behavior of the enterprise business (business problem), thus providing multiple views. This paper discusses a combination approach and investigates the benefits via an illustrative example.

Keywords: Enterprise ontology, DEMO, optimization modeling, linear programming.

1 Introduction

The business challenges of enterprises (companies or firms) in the present economy require new and rich approaches in addressing the issues of optimized production and resource allocation, which cannot be achieved by a single method or approach. It has been evident that a holistic approach, based on combination of methods, allows capturing multiple views, thus, allowing more accurate solution process. In the 21st century economy, where marginal benefits and the trust and confidence of customers are crucial success factors, the stakes are even higher from making inaccurate forecasts or decisions based on simplified outcomes.

The current economic crisis threatens to set back trust and attention between customers and suppliers. As a result, suppliers cannot determine stable sales forecasts. This is a major problem, especially for industries that produce in lots (e.g. pharmaceutical, chemical etc.). Based on these forecasts, planners in such industries need to

J. Barjis (Ed.): EOMAS 2010, LNBIP 63, pp. 128–142, 2010.

determine the optimal product mix to be produced in order to achieve a certain goal (e.g. maximum profit, lowest carbon dioxide pollution) given the constraints of the production facility (e.g. labor costs, machine hours). To solve such questions, planners need to transform the "lot sizing and scheduling" business problem into a mathematical model before they can make use of automated linear programming to determine its optimum. Transforming a business problem into a mathematical model is a crucial activity in business optimization, therefore also of interest in operational research (OR).

As affirmed by Stafford Beer in 1966 [1]: “In operational research model building is conscious”, referring to the importance of good model building. In 1989, Williams [2] recognizes the difficulties for organizations to adopt mathematical model building. The main dilemmas are the perception of the high costs involved with model building and the culture of applying quick-and-dirty rules. Despite these organizational dilemmas, Sarker and Williams [3] argue more rationally why it is difficult to reflect problem aspects mathematically. In many circumstances mathematical model building starts with an unclear problem definition. The Kellogg case [4] of Wolsey and Pochet confirms this conclusion as they found many unstructured problem definitions in minutes of a meeting, internal memos, or reports. Such a practice is considered as a false start for business optimization.

In the literature of operation research (OR), graphical business representations can also be recognized. According to Pinto et al. [5], these graphical representations fulfill two important needs: First, the need to represent a business problem more concisely, and second to support the transformation of a business problem into a mathematical representation. However, graphical business representation divides academia. One line of research argues that OR problems are so diverse that problem-specific representations are required. The other line does not disagree, but believe that problems are sufficiently similar to be treated by the same business-representation approach. Representation techniques in OR, like STN, m-STN, and RTN are scarce in addition they are specific for the problem category of “lot sizing and scheduling” [5]. Although they help to fulfill the need to transform a business problem into a mathematical model, these representation techniques models seem not to serve any another purpose.

In this paper, we apply DEMO (Design & Engineering Methodology for Organizations) as a modeling approach [6] that is applicable for OR for a number of reasons. First, as addressed in the first half of section 3, DEMO is able to represent a business problem in a coherent, comprehensive, consistent, concise, and essential way [7]. Second, as discussed in the second half of section 3, DEMO models are able to capture sufficient essential information to transform a business problem into a mathematical model for optimization. Third, as shown in section 4, DEMO increases the interpretation of the optimization results at hand of a sensitivity analysis especially for decision makers. Finally, in section 5, we argue on the basis of the viable system model of Stafford Beer why interpreting OR results on the basis of DEMO models not only support operational decision making but also strategic ones. To set the stage for our research, we introduce in section 2 all techniques and definitions as applied in this paper. Section 6 contains conclusions and reflections on the findings.

2 Underlying Theories of DEMO and Optimization Modeling

In this section, we first introduce DEMO, which has proven to be an effective approach for capturing the essence of an organizational reality. DEMO models provide a comprehensive essential picture of the organization's business processes that can serve various purposes, including (re)engineering the business processes. In order to apply DEMO models for decision making, performance analysis or forecasting, certain complementary methods (or approaches) ought to be applied that provide more quantitative analysis and numerical results. Such analyses rely on mathematical models, which are applied in optimization modeling based on linear programming and simulation [8]. We introduce a reconstruction of their definitions, structure and process definition which we apply further in this publication.

2.1 Summary of DEMO

DEMO is built on the PSI-theory (Performance in Social Interaction) [9]. In this theory, an enterprise (organization) is a system in the category of social systems [10]. The distinctive property of social systems is that the active elements are human beings or subjects. These subjects perform two kinds of acts: *production* acts (P-acts for short) and *coordination* acts (C-acts for short). By performing P-acts the subjects contribute to bringing about the goods or services that are delivered to the environment. By performing C-acts subjects enter into and comply with commitments towards each other regarding the performance of P-acts. Examples of C-acts are "request", "promise" and "decline". The effect of performing a C-act is that both the performer and the addressee of the act get involved in commitments regarding the bringing about of the corresponding P-act.

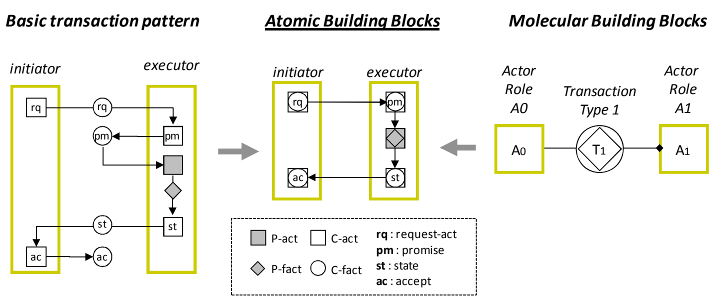


Fig. 1. Ontological building blocks of an organization

C-acts and P-acts appear to occur as steps in a generic coordination pattern, called *transaction*. Fig. 1 exhibits the basic transaction pattern (left), as the elaboration and formalization of the workflow loop as proposed in [11]. A transaction evolves in three phases: the order phase (O-phase for short), the execution phase (E-phase for short), and the result phase (R-phase for short). In the order phase, the initiator and the executor negotiate for achieving consensus about the P-fact that the executor is going to bring about. The main C-acts in the O-phase are the request and the promise. In the execution phase, the P-fact is brought about by the executor. In the result phase, the initiator and the executor negotiate for achieving consensus about the P-fact that is actually produced (which may differ from the requested one). The main C-acts in the R-phase are the state and the corresponding accept. The terms "initiator" and "executor" replace the more colloquial terms "customer" and "producer". Moreover, they refer to actor roles instead of subjects. An *actor role* is defined as the authority and responsibility to be the executor of a transaction type. Actor roles are fulfilled by subjects, such that an actor role may be fulfilled by several subjects and a subject may fulfill several actor roles.

The actual course of a transaction may be much more extensive than the basic pattern in Fig. 1 This is accommodated in the PSI-theory by appropriate extensions of the basic pattern. In the middle of Fig. 1, a comprised notation is shown of the basic transaction pattern. A C-act and its resulting C-fact are represented by one, composite, symbol; the same holds for the P-act and the P-fact. At the right hand side the complete transaction pattern is represented by only one symbol, called the transaction symbol; it consists of a diamond (representing production) embedded in a disk (representing coordination). Transaction types and actor roles are the *molecular* building blocks of business processes and organizations, the transaction steps being the *atomic* building blocks.

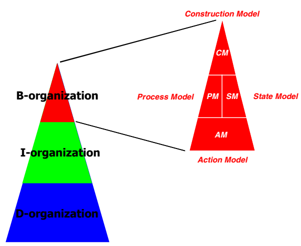


Fig. 2. The three aspect organizations

Another important component of the the PSI -theory is the distinction between three human abilities, which are exerted both in C-acts and in P-acts: the forma, the informa, and the performa ability. Regarding coordination, the forma ability concerns uttering and perceiving written or spoken sentences, the informa ability concerns formulating thoughts and educing them from perceived sentences, and the performa ability concerns being engaged in commitments. On the production side, the forma ability concerns datalogical production (storing, transmitting, copying etc. of data), the informa ability concerns infological production (computing, reasoning), and the performa ability concerns bringing about original new facts (deciding, judging, creating); we therefore call it ontological production.

The distinction between the three human capabilities on the production side gives rise to the distinction of three layered aspect organizations, as depicted in Fig. 2. By definition, the ontological model of an enterprise is the ontological model of its B-organization. DEMO helps in 'discovering' an enterprise's ontological model, basically by re-engineering from its implementation, as e.g. contained in a narrative description. The complete ontological model of an enterprise consists of four aspect models (see Fig. 2). The Construction Model contains the actor roles and transaction kinds, the

Process Model contains the business processes and business events, the State Model contains the business objects and business facts, and the Action Model contains the business rules.

2.2 Optimization Modeling

An enterprise is a complex social-technical system that needs to cope with properties such as uncertainties, agility, and dynamicity. Its performance must be continuously optimized and improved with the respect to the utilization of scarce resources. Optimization techniques (e.g. linear programming, simulation) have been beneficial in many domains establishing their viability and capability as tools, techniques, and methods for decision support. Such techniques include mathematical model building which in fact is a function of a number of variables, subject to certain constraints. Problems that seek to maximize or minimize a mathematical function may be called optimization problems. Many real-world and theoretical problems can be modeled in this general framework. In order to position our proposal to combine linear programming with DEMO, rigorous definitions of the optimization process and its components are required. Fig. 3 presents a general structure of the optimization process, followed by definitions of its components.

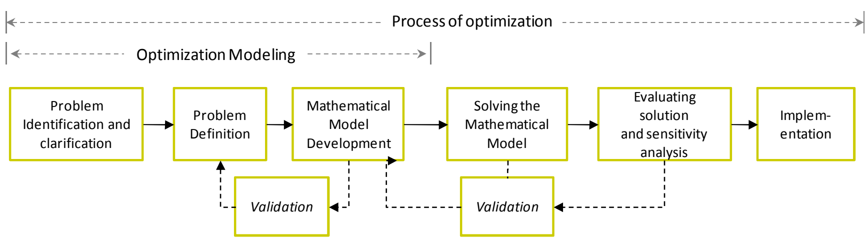


Fig. 3. The process of optimization

Table 1. Definitions in Optimization

Definitions in optimization	
Process of optimization	The purpose of the optimization process is to help determine realistic and practical outcomes of management decision making and design processes. It spans activities as reflected in figure (above)
Optimization modeling	The basic concept of optimization modeling is the process to transform a business problem into mathematical model for the purpose of optimization.
Model	A model is an idealized representation of something
Mathematical model	A mathematical model is an idealized representation of, for example, a business problem that is expressed in terms of mathematical symbols and expressions
Objective function	An objective function is a Mathematical expression in a model that gives the measure of performance for a problem in terms of decision variables.
Decision variable	A decision variable is an algebraic variable that represents a quantifiable decision to be made.
Constraint	A constraint is an inequality or equation in a mathematical model that expresses some restrictions on the values that can be assigned to decision variables.
Parameter	Parameters are constants in the functional constraints and objective function which represents a quantity that is important for the analyses of the problem.
Optimization	The basic concept of optimization is to find the best possible value for a decision variable to a given mathematical model.

Table 2. Building Blocks of a Mathematical Model

Goal Function:	$f(X)$
Constraints:	$g_i(X) \le gb_i$, $i=1,\ldots,m$ $h_i(X) \le hb_j$ $j=1,\ldots,p$ $x_n \ge 0$
$X = (x_1, x_2, \ldots, x_n)$	

The goal function *f(X)* is a function of a set of variables X , see Table 2. The constraint function g_i and h_j are general functions of the variable. The so called right hand sides, gb_i and hb_j are usually the known constants (constraints) to deterministic problems. The nonnegative constraint is added for practical reasons as many parameters cannot be negative. All building blocks together make up the mathematical model. A more practical representation of the mathematical model is presented in the next mathematical representation which is based on the assumption the goal function *f(X)* and constraints $g_i(X)$ are linear functions.

Table 3. Mathematical Model at hand of linear functions

$f(X) = c_1x_1 + c_2x_2 + \ldots + c_nx_n$
$g_1(X) = a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n \le gb_1$ $g_2(X) = a_{21}x_1 + a_{22}x_2 + \ldots + a_{2n}x_n \le gb_2$:

Before we explain the mathematical model representation of Table 3, we present first a set of characteristics of a mathematical model as originally defined by George Dantzig. First, variables reflect resources in an organization to be used for some activity, usually reflected as decision variables. Second, limited quantities of resources are described by parameters. Third, each activity in which resources are used yields a return in terms of the stated goal function. Fourth, allocation of resources is usually restricted by several limitations, known as constraints. Using these characteristics to explain the mathematical model representation of Table 3 leads to the following interpretation:

- the constraint, $g_1(X)$, a_{11} is the resource required from gb_1 for each unit of activity x_1, a_{12} is the resource required from gb_1 for each unit of activity x_2 and so on.
- In the goal function *f(X)*, c_1 is the return per unit of activity x_1, c_2 for activity x_2, and so on.

So, c_i and a_{in} are known as the coefficients of the goal- and constraint function. Our case, which we present in the next section, is compliant to this model so that it exemplifies this rather abstract introduction of a mathematical model.

3 Case Study: Lot Sizing and Scheduling

The problem of lot sizing and scheduling is an area of active research starting from a seminal paper of Wagner and Whithin in 1958. The problem is defined as the challenge of an organization to transform the customer demand into production orders (lots) so as to minimize costs for setups or product change-overs and inventory.

The problem of lot sizing, i.e. transforming customer demand into production orders (lots) so as to minimize costs for setups or product change-overs and inventory, has been a topic of intensive research over the last decades [16]. A realistic lot-sizing model should consider a finite production capacity, several products sharing this capacity, and a dynamic planning situation, which starts from a given production and inventory status and involves a finite planning interval with fluctuating demands. To explore and exemplify the role of DEMO in optimization modeling for such cases we introduce the case of a production plant in pharmacy for which we were able to construct a DEMO model.

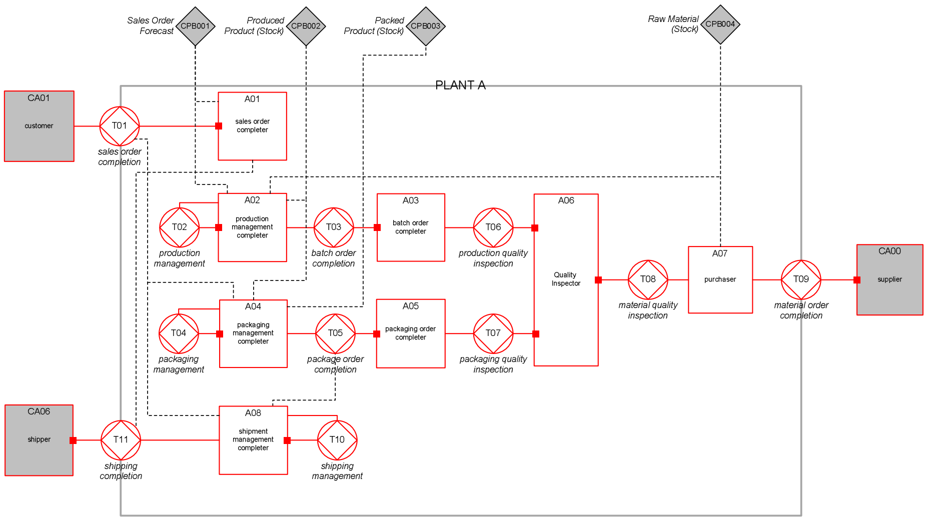
Pharmacy and co. is a global player in pharmacy. A plant in England produces two types of medicines "anti fluenza tablets"(AFT) and "birth control tablets" (BCT) sharing one production line. Their production involves: preparation of ingredients, chemical processing, and making tablets.

3.1 Problem Identification and Clarification

Lot sizing and scheduling is a complex activity. Exploring its literature, a reader will find many examples of lot-sizing and scheduling problems and the manner in which they are treated by operational researchers and computer scientists [12]. It is scientifically justified to conclude that mathematical models are preferred to express a problem in OR. They offer however poor support in the problem definition discussion as mathematical models are 'inconvenient' in the sense of 'difficult to understand'. This justifies the introduction of a conceptual modeling approach to understand the constructional parts of the organization in which this problem exists. We propose to introduce a DEMO model of Pharmacy & Co. for such a role. In the so-called Organization Construction Diagram (OCD) (Fig. 4, top) we see the molecular building blocks of the production plant. Together with the Transaction Result Table (Fig. 4, bottom) it constitutes the ontological model of the organization.

The ontological model includes the transactions of sales, production, packaging, and shipment, including additional transactions for quality assessing and purchase which makes it a realistic and essential lot-sizing model. Both the AF and BC tablets are produced by A03 *"batch order completer"* which capacity is shared for production of two products. A02 *"production management completer"* is responsible to start T03 *"batch order production"* for which he conducts the self initiating transaction T02 *"production management"*. In other words, A02 formally grants a schedule to be executed in T03. We emphasize the complexity of T02, as the optimal mix to be executed in T03 is to be determined in an infological transaction (not a part of this model) based on information about given production-, inventory levels and demand in a rolling horizon.

In this paper we do not use the whole DEMO model, instead we focus on T02 and T03 (dashed border in the transaction result table of Fig. 4 and refined in Fig. 5) to define the optimization problem in more detail.



Transaction	*Transaction Result*
T01 sales order completion	*R01 sales order SO has been completed*
T02 production management	***R02 production management for period P has been done***
T03 batch order production	***R03 batch order BO has been produced***
T04 packaging management	*R04 packaging management for period P has been done*
T05 package order completion	*R05 package order PO has been completed*
T06 production quality inspection	*R06 the quality of batch order BO has been inspected*
T07 packaging quality inspection	*R07 the quality packaging order BO has been inspected*
T08 material quality inspection	*R08 the quality of Supplier Order SUPO has been inspected*
T09 supply order completion	*R09 supply order SUPO has been delivered*
T10 shipping management	*R10 shipping management for period P has been done*
T11 shipping completion	*R11 sales order SO has been shipped*

Fig. 4. Ontological model of Pharmacy & Co.

3.2 Problem Definition

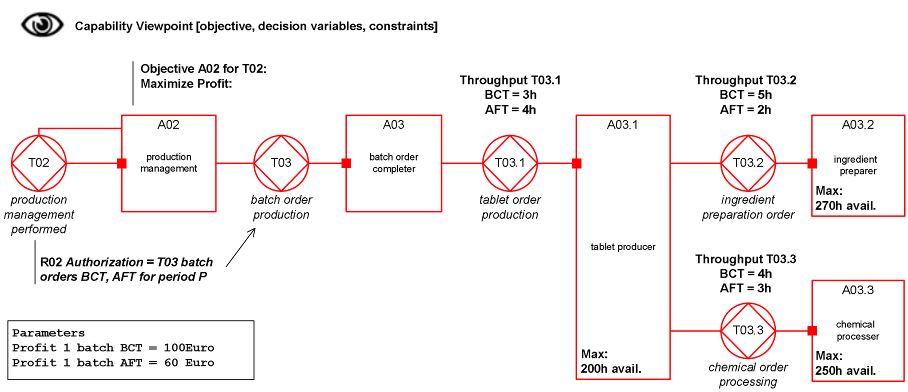
The DEMO model itself does describe the entirety of the ‘Lot Sizing and Scheduling’ problem; it presents ‘only’ the essential structure to which it relates. Aspects need to be introduced. An aspect is a special way to see the so-called 'object of reference' (e.g. the elements of a DEMO representation) which itself is mathematically undefined [13]. In other words the molecular building blocks of transactions have no mathematical meaning, only aspects can have meaning. The aspect element is therefore an abstract concept of a part of the problem. Consequently it has to be operationalised in some way when integrating it into a DEMO model to make it applicable for OR. Reasoned from the structure of a mathematical model, aspects of: (a) the objective to achieve, (b) the constraints to regard, (c) the decision variables and (d) the relevant parameters seem relevant aspects, but does it make sense to combine them with DEMO?. This rather theoretical consideration seems to be confirmed in the way of working of defense organizations captured in the USA Department of Defense Architecture Framework

(DoDAF) and the UK Ministry of Defense Architecture Framework (MODAF). Both frameworks use the same way to define optimization problems, namely based on the capability viewpoint. A capability viewpoint is also applied as a perspective on conceptual models which includes the following elements: a scenario, an objective, the authorizations required, and the existing limitations, see Table 4. These elements map to our definitions of operations research (e.g. decision variable, constraint, and objective) which is not to be expected, as one of the three founding fathers of linear programming, George Dantzig, started optimization during the Second World War for the military.

Table 4. Capability Viewpoint: Lot and Schedule Optimization Q3 2009 Pharmacy &Co.

Scenario	– *Market demand exposed by Customer CA01 is bigger than production capacity.* – *Profit made by producing one lot of BCT is €100 and AFT €60* – *Throughput for BCT for T03.2 5 hours, T03.2 4 hours and T03.1 3hours* – *Throughput for AFT for T03.2 2 hours, T03.2 3 hours and T03.1 4 hours*
Objective	*Maximize profit for Q3 2009*
Authorization	*R02, A schedule plan for Q3 2009 lots BCT and lots AFT to be produced in T03.*
Limitations	*Pharmacy & Co has for Q3 2009, capacity for 270 hours for A03.2 ingredient preparation, 250 hours for A03.3 chemical processing and 200 hours for A03.1 Tablet production.*

We propose the application of the capability viewpoint in DEMO in order to achieve a view which results in a deep and, comprehensive, consistent, concise, and essential (C4E) understanding of the optimization problem. Our approach is applied in the case of Pharmacy & Co. and its result presented in Fig. 5.



Transaction	*Transaction Result*
T02 production management	*R02 production management for period P has been done*
T03 batch order production	*R03 batch order BO has been produced*
T03.1 tablet order production	*R03.1 tabletting order TO has been produced*
T03.2 ingredient preparation order	*R03.2 preparation order PO has been produced*
T03.3 chemical order processing	*R03.3 chemical order CO has been produced*

Fig. 5. Capability View of Pharmacy & Co. (Q3 2009)

The result is not just a diagram. First, a clear set of aspect types (objective, constraints, decision variable and parameters) are defined as a coherent viewpoint for conceptual models. Second, the application of the capability viewpoint in DEMO results in a C4E understanding about the relationships between all optimization aspects. Third, stakeholders are confronted with all the information of a business problem in one artifact. The model reflects the undesirability, complexity, and solvability of a business problem at hand. In conclusion, a capability view, constructed with DEMO and the capability viewpoint, creates a high-definition representation of a business problem at hand. Such an artifact allows an interpretation of the problem at the level of the atomic building blocks of the organization (Fig. 1).

3.3 Mathematical Model Development

As elaborated in the introduction, at hand of Beer [1], Williams [2], Sarker [3], it is difficult for organizations to adopt mathematical model building. A capability view, such as represented in Fig. 5 offers comfort as it clarifies the ontological relationships between all capability aspects. If we regard the throughput parameters, we observe their relationship is defined by the ontological relationship between transactions T03.1, T03.2, and T03.3. This offers a new perspective on mathematical model building. A new way of working in mathematical model building, based on following the ontological relationships between the capability aspects, is now possible. The results of such a way of working are presented in Table 5.

Table 5. Mathematical Model of Pharma & Co.

Decision Variables for A02	BCT_{T03} = *lots of BCT to be scheduled for T03 in Q3 2009* AFT_{T03} = *lots of AFT to be scheduled for T03 in Q3 2009*								
Objective Function	*Maximize* $Profit_{R03} = 100BCT_{R03} + 60AFT_{R03}$								
Constraints	***3h***	$T03.1_{BCT}$	+	***4h***	$T03.1_{AFT}$	≤	***200h***	*Cap A03.1*
	5h	$T03.2_{BCT}$	+	***2h***	$T03.2_{AFT}$	≤	***270h***	*Cap A03.2*
	4h	$T03.3_{BCT}$	+	***3h***	$T03.3_{AFT}$	≤	***250h***	*Cap A03.3*

We observed we applied some general assumptions during the application of the capability viewpoint in DEMO, namely:

a) Decision variables are ‘owned’ by a DEMO actor (usually in an initiator role).
b) Resources can only be an aspect of a DEMO actor (usually in an executor role) and is measured by a common unit (e.g. hours or utility).
c) The goal function is constructed at hand of a set of DEMO transaction results.
d) Parameters can be an aspect of any DEMO element.

Based on these assumptions we have an indication of a consistent relationship between the elements of the mathematical model (Table 3) and DEMO elements (Fig. 1). For instance the linear programming definition of “unit of activity x_1” maps to a DEMO transaction. Although all definitions can somehow be mapped it is too early to speak of a coherent and consistent relationship between all elements of both models. In the case of Pharmacy & Co. a level of consistency between both models could be observed but we realize that sometimes it is difficult to allocate an aspect uniquely to an element of a DEMO model.

4 Interpreting Optimization Results with DEMO

In the previous chapter we introduced the role of DEMO in optimization modeling. In this section we explore its role in the other phases of the optimization process (Fig. 3). If we look beyond the scope of optimization modeling, a new challenge of solving the mathematical model and interpreting its results arises. Generally the simplex algorithm is applied to solve a mathematical model. The produced results are very abstract because usually it is presented as a mathematical matrix (final tableau). Within this section we present optimization results for Pharmacy & Co based on its DEMO model. We explore the functionality of the DEMO model and emphasize its meaning for decision makers in interpreting of the optimization results.

4.1 Solving the Model

Our results are based on Lindo, which is a free software program to solve linear programming problems based on the simplex algorithm. The ideal product mix is to produce 48 (48.57143) Lots of BCT and 13 (13.57143) Lots of AFT in T03. A profit of € 5671 (5671.429) would be realized if T03 would be initiated according this mix. We will not elaborate further in the practical usage of the Lindo as we are only interested in the results and not in a particular tool. However, we like to mention that these results can also be achieved with other linear programming software available on the market today.

4.2 Sensitivity Analysis in DEMO

The sensitivity report has two parts, 'the final tableau' and 'the allowable increase and decrease report'. New terms need to be introduced in order to understand them. The final tableau represents the reduced cost or shadow price. A shadow price is the change in the objective value obtained by relaxing a constraint by one unit. The allowable increase is the amount by which the coefficient of the objective function can be increased without causing the optimal basis to change. The allowable decrease is the amount by which you the coefficient of the objective function can be decreased without causing the optimal basis to change. All definitions and their meaning will be exemplified for Pharmacy & Co.

Table 6. Final Tableau for 'Capability Viewpoint Q3 2009' of Pharmacy & Co.

	T03 *BCT*	***T03*** *AFT*	***A03.2*** *slack*	***A03.3*** *slack*	***A03.1*** *slack*	***Solution***
Profit	*0.000*	*0.000*	*15.714*	*0.000*	*7.143*	*5671.429*
T03 BCT	*1.000*	*0.000*	*0.286*	*0.000*	*-0.143*	*48.571*
A03.3 $_{Slack}$	*0.000*	*0.000*	*-0.500*	***1.000***	*-0.500*	*15.000*
T03 AFT	*0.000*	*1.000*	*-0.214*	*0.000*	*0.357*	*13.571*

Table 7. Ranges in which the optimum remains unchanged

	Unit of measurement	***Current***	***Allow. Incr.***	***Allow. Decr.***
T03	*Profit on R03 (T=BCT) in Euro*	*100*	*50*	*55*
T03	*Profit on R03 (T=AFT) in Euro*	*60*	*73.3*	*20*
A03.2	*Work Capacity in hours*	*270h*	*30*	*170*
A03.3	*Work Capacity in hours*	*250h*	*INF.*	*15*
A03.1	*Work Capacity in hours*	*200h*	*30*	*38*

Table 6 and Table 7 present the output of the sensitivity analysis as supplied by Lindo. The final tableau (Table 6) is a matrix, which is the result of the simplex algorithm that applies pivoting actions according some optimization objective. This process starts from a first tableau towards a final tableau. A requirement to setup the first tableau is the introduction of slack variables for each constrained resource. A slack variable represents the amount of a constraining resource or item that is unused. The simplex algorithm pivots around the lowest value in the matrix, it stops when the values of the slack variables are minimized. That matrix is the final tableau. During this process ranges in which the optimum changes or remains unchanged are recorded and reflected as a part of the sensitivity analysis, see Table 7.

We present in this section our interpretation of these results from the DEMO ATD (Fig. **6**). Two lines of interpreting will be presented. First, we interpret the shadow price in its constructional context of a DEMO ATD. It provides decision makers powerful insight into the sensitivity of the operational construction. Second, we discuss the ranges in which the optimum remains unchanged.

In the case of Pharmacy & Co. the final tableau (Table 6) contains two shadow prices. The first shadow price is based on the increase of the capacity of A03.1 by 1 hour, the second is based on the increase of the capacity of A03.2 by 1 hour. Fig. **6** presents in 'Arial style' only the second shadow price. The shadow price is in fact a 'what-if' analysis. It starts in this case with increasing the capacity of ingredient

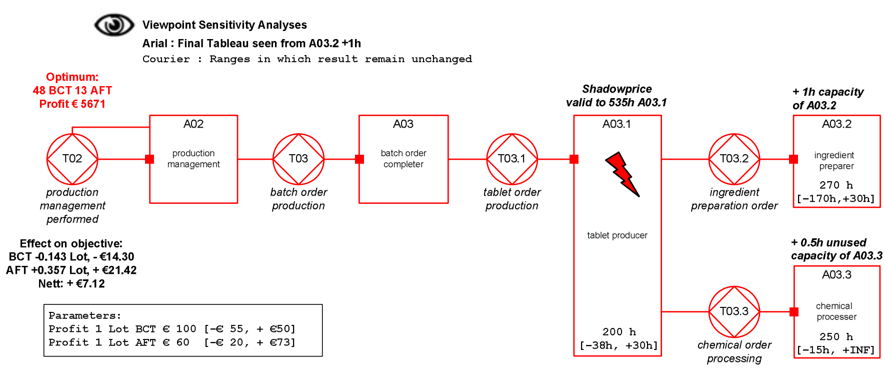


Fig. 6. Sensitivity analysis View on the DEMO Actor Transaction Diagram for Q3 2009

preparation A03.2 by one hour. The effect is: (a) the unused capacity of A03.3 (chemical processing) increases by half an hour, (b) less BCT of -0.143 lot (- €14.30) is produced and (c) more AFT +0.357 lot (+€21.42) is produced. This results in a positive Nett effect of €7.12 on the total profit. So increasing the capacity of A03.2 seems profitable. This has however a limit. Each hour extra A03.2 decreases the production of BCT by -0.143 lot. Producing BCT cannot drop below zero, so to earn an extra €7.12 is limited up to 48/0.143 = 335 hours above existing capacity of A03.1 of 200hours = 535hours. The reason why A03.1 is pointed as the 'bottleneck' is determined by $A03.1_{slack}$ variable in the final tableau, see the dashed column in the final tableau in Table 6.

The allowable increase and decrease is also presented in figure 6 in `'COURIER'` style. We see from this perspective the allowable bandwidth in which parameters and constraints may change, without affecting the optimization results. We identified the next critical results in all parts of the sensitivity analysis of Pharmacy & Co. for Q3:

- The capacity of A03.3 'Chemical processing' may not decrease with 15h
- +1 hour A03.1, +€15.8 profit, up to capacity of 330 h of A03.2

So a hypothetic managerial operational consideration to pay up to 2 euro/h overwork for ingredient preparation is profitable. Another managerial decision is for example to outsource 'ingredient preparation'. The organization and its parameters would change so the mathematical model would be invalid. Such decision making is of another order and sets the stage for a discussion about constructional decision making in respect to operational objectives. We will explore the value of the proposed combinatory approach of DEMO and OR in such a context in Chapter 5.

5 Discussions

As illustrated, the combination and alignment between the mathematical approach (and its derived model) and DEMO approach (and its derived model), realized by allocating the capability aspects in DEMO, connects the two 'Ways of Thinking'. This is due to the differences in the initial purpose of both models. Obviously, the mathematical model represents the operational challenge (behavior), while DEMO represents the essential construction of the organization (structure). So, both models combined deliver a feasible space for decision makers in which constructional options (e.g. in-sourcing, resourcing, outsourcing, mergers and acquisitions) are allowed in respect to operational (behavioral) constraints. With this holistic view of the problem, such a model driven decision support has multiple advantages that we discuss through the case we presented in this paper.

First, outsourcing of ingredient preparation in the case of Pharmacy & Co. is a feasible constructional consideration. It influences the objective of maximizing profit positively, if it is determined based on an unchanged mathematical model. This may not be the case. In our opinion synthesis between DEMO and mathematical representations (or views) needs to be applied so the consequences of a constructional change are reflected in a new mathematical model. Examples of such consequences is to add a new constraint which reflect the initial cost of the constructional change which may not exceed a certain limit, or values of other parameters (e.g. throughput) which may change as result of the constructional change. So, a synthesis between mathematical model and DEMO is of advantage as it increases the understanding of the business problem in OR and Enterprise Engineering.

Second, the clear structure of the business capability viewpoint increases the capturing of business reality and the understanding of the business problem. Furthermore, it separates the concerns of OR and Enterprise Engineering professionals in an approach which remains combinatory. This increases the dialogue between both disciplines (and their approaches) necessary to study all options in both directions. The capability viewpoint is therefore a crucial technique that binds the theories of OR and DEMO together.

Third, a sharp distinction between operational and strategic decision making can be observed in the proposed approach. This observation was made in the VSM of Stafford Beer. The VSM defines levels of decision making in order for the business system to remain viable. Based on this model, we may conclude that operational decision making is influencing the operation of a system (organization), but leaving its construction unchanged. In DEMO terms, this means executing transactions for allocating resources but also for changing parameters and changing the order in which actors process their to-do list. Strategic decision making is redesigning the construction of a system (organization). In DEMO terms this means even changing the Action Model, so replacing a business rule by another one. In our opinion these definitions have a practical relevance.

6 Conclusions

In this paper, we proposed an approach for combining an OR approach (linear programming) and enterprise engineering approach (DEMO) as the two together yield a more holistic view of the underlying problem – the business problem. The paper, through a 'Lot Sizing and Scheduling' problem, demonstrated advantages of such a combinatory approach as it: (a) is concise in defining a business problem, (b) offers a smooth way to transform a business problem into a mathematical model and (c) increases the interpretation of optimization results (i.e. the results of the sensitivity analysis). Our conclusions are that DEMO helps to model the problem domain at business level based on profound theory. This model can serve as an input for any analytical analysis, in the case of this paper linear programming for its simplicity. Furthermore, DEMO helps to focus on essential and more stable processes and activities, which by itself reduces complexity of the underling problem and helps scoping of the problem. This reduction of complexity and systematic scoping facilitate to design a more accurate formal models using, for example, linear programming. Although in this paper we use linear programming after initially understanding and modeling the business domain, the DEMO models can be used as an input for any other formal method as well. By stating that DEMO helps to obtain more accurate solution of the problem, it is meant that the focus is made on essential activities, i.e., reducing unnecessary complexity, and the model is based on profound theory. These two superiorities result in more accurate solutions when linear programming is used subsequently."

For future scientific research we follow the strategy to invest in the development of the viewpoint concept. This article has shows that a viewpoint functions a mediation concept [14] on which grounds OR approaches and DEMO can be combined. We set two important requirements for such a viewpoint: (a) it needs to lead to a view which present the behavioral aspects (quantitative, mathematical) and social construction (ontological) of a problem and/or feasible solution in one model and (b) these views need to increase the quality of decision making in OR (approaches). Disconnected from each OR approach (e.g. Lean, Six Sigma, Linear programming, simulation etc.) a DEMO model offers already insight in the essential construction of an organization, which itself is a stable perspective. We are aware that this nature changes after a viewpoint is applied. A view, generated by a viewpoint, is a temporal view (e.g. in case of Pharma and Co. limited to Q3) and problem adequate (e.g. to linear programming). The major advantage is however that such a view functions as a problem description still connected to essential construction of an organization. So, applying OR specific viewpoints for DEMO would mean a new way of model driven decision support for operational- and strategic business decision making based on a positive business case. However, for these advantages to be harvested, deeper research needs to be conducted. In particular, guidelines and criteria allowing when and how DEMO should be used, and what additional insights DEMO bring to the study of an OR problem, need to be developed.

References

1. Beer, S.: Decision and Control: The Meaning of Operational Research and Management Cybernetics. Classic Beer Series. Wiley, Chichester (1994)
2. Williams, H.P.: How Important are Models to Operational Research? IMA J. Management Math. 2(2), 189–195 (1989)
3. Sarker, R.A., Newton, C.S.: Optimization Modelling: A practical Approach. Manufactoring & Industrial Engineering. CRC Press, New York (2008)
4. Pochet, Y., Wolsey, L.A.: Mixed Integer Programming Algorithms and Decomposition Approaches. In: Production Planning by Mixed Integer Programming, pp. 185–206 (2006)
5. Pinto, T.N., BarboÌsa-PoÌvoa, A., Novais, A.: Design of Multipurpose Batch Plants: A Comparative Analysis between the STN, m-STN, and RTN Representations and Formulations. Industrial & Engineering Chemistry Research 47(16), 6025–6044 (2008)
6. Dietz, J.L.G.: DEMO: Towards a discipline of organisation engineering. European Journal of Operational Research (2001)
7. Dietz, J.L.G., Hoogervorst, J.A.P.: Enterprise ontology in enterprise engineering. In: Proceedings of the ACM Symposium on Applied Computing, pp. 572–579 (2008)
8. Barjis, J.: A Business Process Modeling and Simulation Method Using DEMO. In: Enterprise Information Systems, pp. 254–265 (2008)
9. Dietz, J.L.G.: Enterprise Ontology. Enterprise Ontology (2006)
10. Bunge, M.: Treatise on Basic Philosophy. Ontology II: A World of Systems, vol. 4. Springer, Heidelberg (1979)
11. Denning, P., Medina-Mora, R.: Completing the loops. In: ORSA/TIMS Interfaces (1995)
12. Staggemeier, A.T., Clark, A.R.: A survey of lot-sizing and scheduling models. In: 23rd Annual Symposium of the Brazilian Operational Research Society (SOBRAPO), pp. 938–947. Campos do Jordao SP, Brazil (2001)
13. Dresbach, S.: Modeling by Construction: A New Methodology for Constructing Models for Decision Support. In: Hawaii International Conference on System Sciences, p. 178 (1996)
14. Töpfer, A.: Erfolgreich Forschen, Ein Leitfaden fuehr Bachelor-, Master-Studierende und Doktoranden. Springer, Lehrbuch (2009)

User Interfaces Modelling of Workflow Information Systems

Wided Bouchelligua[1], Adel Mahfoudhi[2], Nesrine Mezhoudi[3], Olfa Daassi[3], and Mourad Abed[1]

[1] LAMIH, University of Valenciennes UMR CNRS 8530, BP: 311-59304 Valenciennes cedex 9, France
{wided.bouchelligua,mourad.abed}@univ-valenciennes.fr

[2] CES, ENIS Soukra Road km 3,5, B.P: w 3038 Sfax TUNISIA
adel.mahfoudhi@fss.rnu.tn

[3] UTIC, ESST Taha Houssein Road 5, B.P: 56, Bab Menara, 1008 Tunis TUNISIA
nesrine.mezhoudi@yahoo.fr, olfa_daassi@yahoo.com

Abstract. In the recent years, the workflow system has increasingly gained considerable attention in the Information Systems community. It allows the integration of the various users to reach the objectives of the organization. The users operate their Information Systems through the User Interfaces (UI). Therefore there is a need to take into account the workflow model in the development process of the UI for an Information System. This paper has two objectives to attain. The first one is the proposition of a model driven approach to derive a plastic UIs of a Workflow Information System. The second objective lies in the use of Business Process Modelling Notation (BPMN) for the modelling of the interaction models (Task Model, Abstract User Interface, Concrete User Interface) with the aim of supplying more adapted models for a business expert who represents a main candidate in the success of the organization. It is through a case study applied to a real information system that our approach proves to be reliable.

Keywords: User Interface, Model Driven Approach, Workflow Information System, Plastic User interface, Business Process Modelling Notation.

1 Introduction

Nowadays, Business Process (BP) is increasingly holding a big interest in the field of Information Systems (IS). According to [9], Business Process is a structured set of activities ordered across time and space to produce a specific result for a particular customer or market. In addition, Workflow systems are defined by Workflow Management Coalition (WfMC) as the automation of a business process during which documents, information or tasks pass from one participant to another for action according to a set of procedural rules [36]. Therefore, Workflow systems help the organization to guarantee the regular adaptation of

J. Barjis (Ed.): EOMAS 2010, LNBIP 63, pp. 143–163, 2010.

its BP to its permanently evolving environment [12]. However, the adaptation of the User Interfaces (UI), through which the users process their IS, is often neglected due to the change of the organization context.

In brief, not only do interfaces have to answer the new requirements listed at the level of the workflow of an information system, but also to have the capacity to adapt themselves to the changes of their context of use. This type of interface is said to be plastic or sensitive to the context. The plasticity was defined for the first time by Thenvenin [32] as the capacity of a user interface to adapt to the context of use while preserving usability. The context of use is denoted by the triplet <user, platform, environment>. In this area of research, we can quote the TERESA method [23] which supplies a single model, that of the tasks, and allows the generation of several interfaces for various platforms. We also quote the Comets (COntext sensitive Multi-target widgETS) [8], which proposes essentially a model for the plastic interactors which are capable of adapting themselves to the context of use. To have plastic user interfaces for an IS we, firstly, opt for the integration of the workflow model as the initial model in an approach based on a set of models. Secondly, we consider the variant of the interaction platform in order to propose the adaptation rules.

The proposed approach benefits from the advantages of the domain of the Model Driven Engineering (MDE) [10]. The latter goes beyond the framework of Model Driven Architecture (MDA) [21], which can be summarised in the elaboration of the platform independent models and in their transformation into platform specific models [2], to cover the methodological aspects.

The remainder of this paper is structured as follows: section 2 gives fundamental concepts in MDE and how they can be applied in Human Computer Interface (HCI) domain. Section 3 briefly reviews the model based approaches for the development of workflow user interfaces. Section 4 describes the proposed approach to derive UIs through a case study. Finally in section 5 we draw the conclusion and provide perspectives to future research.

2 Model Driven Engineering Concepts

Since the recent adoption of the MDA by the OMG [24], the model driven approach has aroused a big interest. Then, the MDA approach has become a particular variant of the Model Driven Engineering (MDE) to cover the methodological aspects as well.

The MDE is based on three essential concepts: the models, the meta-models [31] and the transformations. These frequently used terms in the MDE and the relations between them were widely discussed in the literature [2], [3], [10], [11] and [17]. In [3] Bézivin identifies two fundamental relations: the first relation called **RepresentedBy** is connected to the notion of model, and the second called **ConformsTo** defines the notion of model with regard to that of meta-model (Fig. 1).

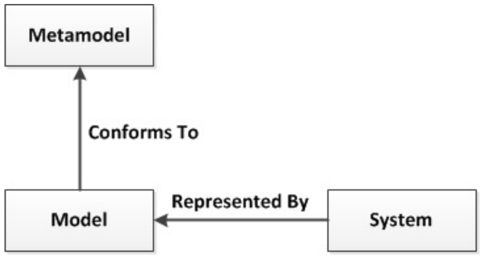


Fig. 1. Basic Notions in Model Driven Engineering

Although there are many definitions for the model concept in the literature, there is a convergence between them. Actually, they all aim at making reference to the notion of model and modelled system. Indeed, an aspect of a system is captured by a model which is linked to a meta-model in a relation called RepresentatedBy and noted μ. A meta-model is a model of a modelling language, which leads to the identification of a second relation named ConformsTo [3] [11]. Such a relation, noted by χ allows assuring the productivity of a model because it is in compliance with its meta-model. This facilitates the transformation of models. The notion of transformation is another central concept for the MDE, the mechanism of transformation allows using both Model and Meta-model notions. The power of the MDE consists in creating the transformation models, which build on meta-model corresponding to the source model and the target model. So the noted τ relation IsTransformedInto allows the automation of the transformation of a model into another.

In this section, we have chosen to present the various concepts and relations of the MDE, to show their correspondences in the field of the HCI. The purpose of our work is to benefit from MDE techniques in an approach of generating the UIs with models. The idea for the HCI models is to migrate from a conceptual to a productive aspect. The marriage of two domains turned out to be very promising, that is why so much work focused on it. The initiative of this area of research is founded by [26].

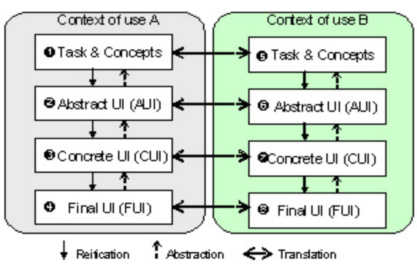


Fig. 2. Cameleon Reference Framework [35]

The work of Sottet [26] is among the first one to have joined the Model Driven Engineering and the domain of the Human Computer Interaction. His approach makes it possible to show that the concepts of the MDE could be usefully applied to the engineering of the UI. Sottet [26] proposes meta-models and transformations of models to divert plastic UI. Indeed, the Cameleon reference framework [7] defines four essential stages of user development of the user interfaces in a pervasive environment (Fig. 2): tasks and concepts, abstract interface, concrete interface, and final interface.

3 Background and Related Work

The integration of the workflow model into their approaches to derive the UI has been a recent field of research. We can, firstly, refer to the work of [22] which has demonstrated the importance to incorporate the conception of the graphic interfaces at the time of the business modelling of the application, and thus representing a first sketch of the link between the graphic interfaces and the business models. Besides, we can consult the work of [18] or even that of [29] which propose improvements to the modelling by the business process for the purpose of integrating it to the graphic interfaces. Moreover, [30] suggests an approach of development of the workflow interactive applications, which support a quick creation of initial workflow system and a derivation of the adaptable UI. Furthermore, the work of Kristiansen [19] shows the importance to use both models (task model and workflow model) in an approach of role-oriented conception of UI. She gives evidence that: "The workflow model defines what tasks need to be fulfilled and their possible ordering; hence the workflow model is suitable as a "frame" for creating task models" [19]. In the same vein of thought, Guerrero [15] confirms that "The workflow model defines what processes and tasks need to be fulfilled and their possible ordering; hence the workflow model is a "framework" for creating task model ...". In addition, [15] proposes an approach of engineering managed by the models to derive the user interfaces from the workflow model and the task model. His methodology builds on a conceptual model composed of: workflow, process, task and organizational structure. Unlike the aforementioned approaches, the recent work of Traetteberg [33] considers only one model: workflow model which is an oriented task. This model is the building stone of the conception of the dialogue model.

Based on the same fundamentals of the approaches proposed by [19] and [15], whose framework for the creation of the task model is the workflow Model, we envisage the proposition of a new approach to derive a plastic UI of a workflow information system. Besides, we opt for an easy to use notation by the domain expert who is the main candidate in the success of the enterprise.

4 Proposed Method

Aiming at illustrating the different models and meta-models, transformation rules and adaptation rules of the proposed approach, we have chosen a case study applied to a real information system. The data-processing service company "Himilco Electronic banking" specialized in the implementation of banking solution and security of transactions and payment (e-payment) tend to conceive an application which allows the normalization of the possession process of a credit card. The scenario of the system is extracted from [1].

The workflow of this process has to satisfy a set of requirements which can be summarised in:

– Update of the rules base and the criteria of credit cards by the responsible service.
– The customer's command of a credit card from the bank web site.
– Reception of the command by the customer responsible for the check of the conformity of the data.
– Evaluation of the request by the analyst.
– The application has to satisfy all these requirements as well as those of the authentication of every system user, especially that the application is intended for banks.

In what follows, we, firstly, define a set of models and meta-models to which they are conform. Secondly, we present a description of all the transformations and we explain the major principles of UI adaptation based on a platform meta-model.

4.1 Models and Meta-Models of Method

Our approach follows the steps of the expanded Cameleon framework [16]. The extension involves the first level of abstraction (Task&Concepts). In this stage, the task model is evolved to back up the interfaces modelling and development for a workflow information system. To reach this objective, Guerrero considers a workflow model decomposed into processes that are in turn decomposed into tasks [16]. A conceptual model composed of workflow, processes, tasks and organizational units, is proposed to support this stage. This model is given in details in [16]. The workflow model describes the flow of work between the organizational units that represent the users and their business roles. Guerrero uses Petri nets [25] to describe this model. Besides, the process model shows the arrangement of the tasks from the time, space and resource perspectives (human, material and immaterial). Furthermore, the task model characterizes the division of tasks into sub-tasks connected by temporal operators from a user point of view. In [16], ConcurTaskTree (CTT) notation [23] is used to describe the task model. In short, this stage consists in the elaboration of three models: the Concept Model, the Workflow Model and the Task Models associated with the processes forming the Workflow model.

Workflow Model. Our Workflow Model is based on the Business Process Modelling Notation (BPMN) [5]. It is based on a set of simple graphic elements, easy to use and to bring together, which makes this language simple to treat by business experts. Besides, BPMN is in the course of standardization by the Object

Management Group (OMG) [24]. One of the biggest characteristics of the BPMN notation is that it is built on the Petri networks, which allows the validation of the models. By using the BPMN notation, the workflow is presented in the form of a set of activities that can be processes, sub-processes or elementary tasks. The set of activities is organized in the form of the containers which represent the partitions of the process showing a distribution of the activities by participant (an actor or a particular organizational entity). Fig. 3 shows the workflow model associated with the business process of the possession of a credit card. This business process presents one of several business processes which can be used in a bank. This workflow brings about four actors:

- The customer: a customer of the bank asking for a credit card;
- The customer responsible: the system administrator;
- The service responsible: the bank agent;
- The analyst: the financial analyst of the bank.

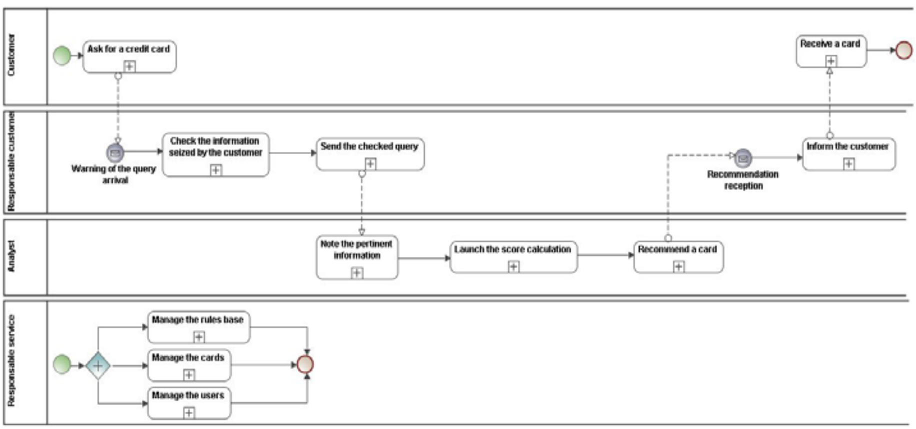


Fig. 3. Workflow Model of the possession process of the credit card

The BPMN model presents the workflow between the various actors of the system. The customer sends a request for a credit card to the customer responsible "Ask for a credit card". This request describes:

- His type (private individual, company),
- His information (Account number, ID card Number, Last name, First name ...),
- And the type of sought card.

The customer responsible verifies the information seized by the customer "Check the information seized by the customer" and sends the verified requests "Send the checked request" to the analyst. The analyst evaluates the request by noting the relevant information “Note the pertinent information” and launching the calculation of the score “Launch the score calculation”. After this evaluation, he recommends a card to the customer “Recommend a card”. The customer responsible receives the recommendation and sends it to the customer “Inform the customer”. The service responsible manages the rules base “Manage the rules base”, the types of cards in the banking institution “Manage cards” and the users “Manage the users”.

Task (Meta-) Model. According to the expanded Cameleon framework [16], each non-extended process contained in the workflow model is considered as a high-level task for a task model in the UI sense. That is why each process gives birth to a task model. The identified task models are going to feed the UI generation process proposed by the Cameleon reference framework.

Having overviewed the state of the art on the task and workflow models, we can conclude that both models share a large number of concepts. Moreover [28] proposes a methodology of alignment of a business process expressed in BPMN towards the models of user interface, more particularly, the construction of a CTT task model from the business process on the basis of a set of rules. Because these rules are associations between the elements of the BPMN business process and those of the CTT task model, reflections were drawn for the description of the task model on the basis of the BPMN notation. Our idea is confirmed by [19]: “Because of the considerable overlap in workflow and task modelling concepts, we have considered the possibility of extending BPMN so that it also can be used for task modelling”. BPMN uses containers (lane) to model the process according to an organizational view. Yet, a task model is built according to a compositional view which leads to this notation extension to be able to apply the task modelling.

Our task model contains a global sub-process presented in the form of a non-extended sub-process called “CollapsedSubProcess”. It should be born in mind that the BPMN notation presents a sub-process in two forms; “non-extended” (CollapsedSubProcess) or “extended” (ExtendedSubProcess). With each non-extended sub-process is associated an extended sub-process where the contents of the sub-process are detailed. This global sub-process is then decomposed in the form of abstraction levels “LevelofDiagram”. A “LevelOfDiagram” presents a task decomposition level. This decomposition is made under hierarchical shape. Each level contains a set of extended sub-process “ExtendedSubProcess” spreading the set of non-extended sub-process “CollapsedSubProcess” of a superior level. Each extended sub-process “ExtendedSubProcess” contains a set of the tasks “Task” and/or non-extended sub-process “CollapsedSubProcess” as well as the relations between them. These relations are modelled by “SequenceFlow”. “SequenceFlow” allows the connection between BPMN elements belonging to “ExtendedSubProcess”. The entry “Gateways” allows the making of conditional connections during the execution of a process or a business sub-process. A set of events of the launching of process “StartEvent” generates a token which will be consumed by a terminal event modelled by “EndEvent”. All the elements of the task model can be decorated through a constituent annotation

"Annotation". These annotations serve to decorate the tasks by the domain concepts. We have established a meta-model for the extended notation. Elements and constraints constituting this meta-model are described by means of associations and cardinalities.

The Task Model is structured in the form of levels (LevelofDiagram). It is built according to the subsequent procedure:

- **Stage 1:** Each non-extended sub-process of the workflow model is represented by a global sub-process of "CollapsedSubProcess" type;
- **Stage 2:** At level 1, specify the non-extended sub-processes and/or the elementary tasks by detailing the global sub-process;
- **Stage 3:** Use the gateways and the types of sequence flow to identify the inter activity relations (a sub-process or task) within the extended sub-process;
- **Stage 4:** Use the event types to specify the start, intermediary and end events within the extended sub-process;
- **Stage 5:** recall stages 2), 3) and 4) to identify all the sub-processes and all the elementary tasks of a sub-process to be found in the superior level.
- The end of the construction procedure of task model is characterized by the acquisition of all the actions.

Thanks to the Graphical Modeling Framework (GMF) tool [13] of Eclipse, we developed a graphic editor for our task model based on the meta-model proposed in Fig. 4. GMF is a framework that allows the generation of graphic editors. It is based on EMF (Eclipse Modeling Framework) and GEF (Graphical Editing Framework). The created model with our realized editor can be serializable as XML, which is compliant with the task meta-model.

The realized tool is a plug-in for the Eclipse platform of development. Fig. 4 presents the task meta-model and the task model associated with the sub-process "Ask for a credit card" realized by the customer. From level 1, we detail the sub-process and the elementary tasks for sub-process "Ask for a credit card". This request consists in logging in, then choosing the customer type (private individual or company), finally determining a form.

Concept (Meta-) Model. Each data listed in the analysis of the requirements is modelled in a Concept Model. Each concept is connected to one or several tasks in which it contributes to their realizations. The domain concepts can be physical or abstract entities from the real world having a final representation at the level of the interface. These concepts intervene in the realization of the user task.

The state of the art proposes diverse formalisms for the representation of Concept Model: the entity-relation model [4] and the UML class diagram [14]. Our method suggests a concepts model based on UML class diagram. Fig. 5 proposes a simplified concept meta-model for an UML class diagram and a concept model illustrated by the case study. For example, the "Customer" represents a "Private Individual" or a "Company". The "Customers Responsible" manages the information connected to one or several customers. The "Responsible Service" manages the weight "WeightField" and the card types of "Card" in the banking institution.

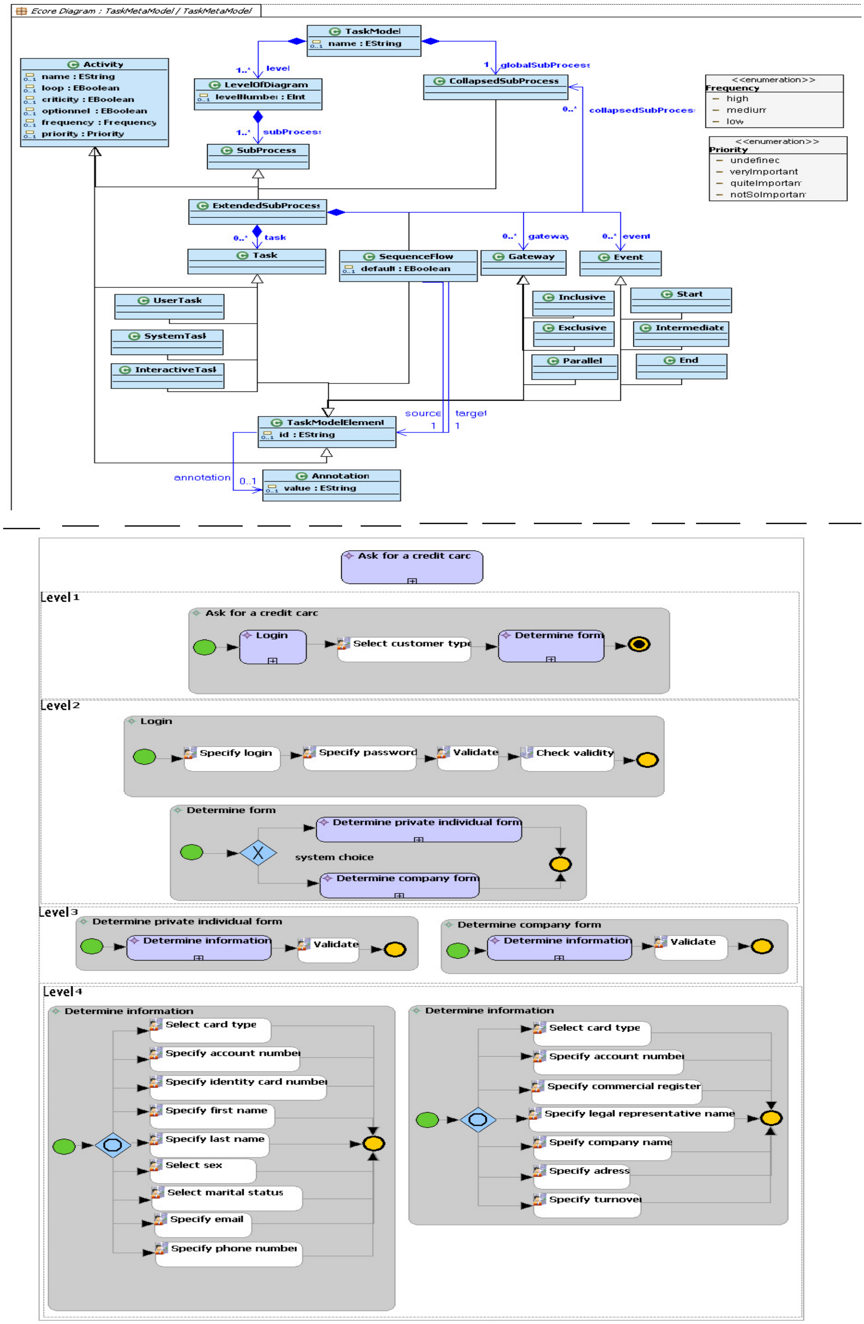


Fig. 4. Task Meta-model and Task Model for the possession process of the credit card

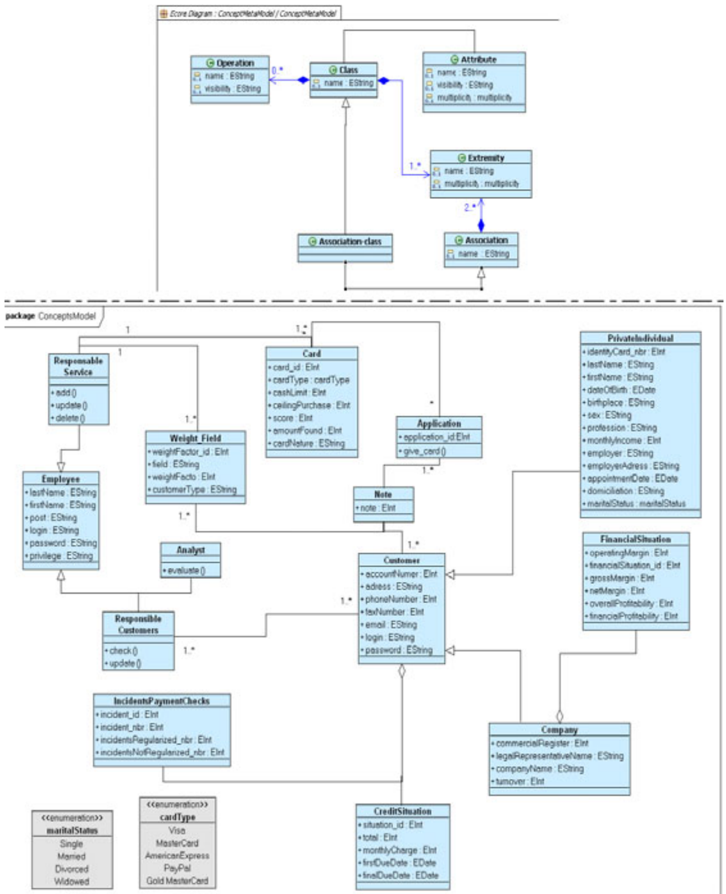


Fig. 5. Concept Meta-model and Concept Model for the possession process of the credit card

Abstract User Interface (Meta-) Model. In the literature, the abstract user interface is defined in several ways. In fact, Thevenin [32] defines it as a set of interconnected workspaces. A workspace is an abstract structure in which an interaction is organized. The connection between workspaces is made according to links between the tasks and the domain concepts.

In our approach, the Abstract User Interface (AUI) allows the transition of the specification in the modelling of the abstract components of the interface. In order to describe the Abstract User Interface and the Concrete User interface, we have appeal to a static model of the interactions [6]. Aiming at applying a model-to-model transformation, we have refined the static model of the interactions of [6] in the form of two meta-models: The AUI and CUI meta-models. AUI meta-model which is shown in [6] describes the hierarchy of the abstract components "UIComponent" corresponding to logical groups of interactions "UISpace". The modelling of the abstract interface of an application is then made by one or several "UIGroup" which model containers forming coherent graphic elements (a window in a Windows environment, for example). Each "UIGroup" consists of one or several "UIUnitSuit" and/or "UIUnit". A "UIUnit" gathers a set of interaction elements which cannot be separated from a logical business point of view of the application (a treatment form for example), a "UIUnit" can include one or several "UISubUnit". The advantage of this modelling is to allow the creation of the application by assembling the existing elements, resulting in a strong reusability. The AUI is expressed by means of the BPMN notation, through the use of a similar formalism, which is in harmony with the established workflow and the task model. Fig. 6 shows also the abstract user interface for the possession process of the credit card. This interface contains a "UIGroup" associated with the global sub-process "Ask for a credit card". This "UIGroup" gives access to two "UIUnitSuit" ("Login" and "Determine private individual form") and "CollapsedUIUnit" ("Select customer type").

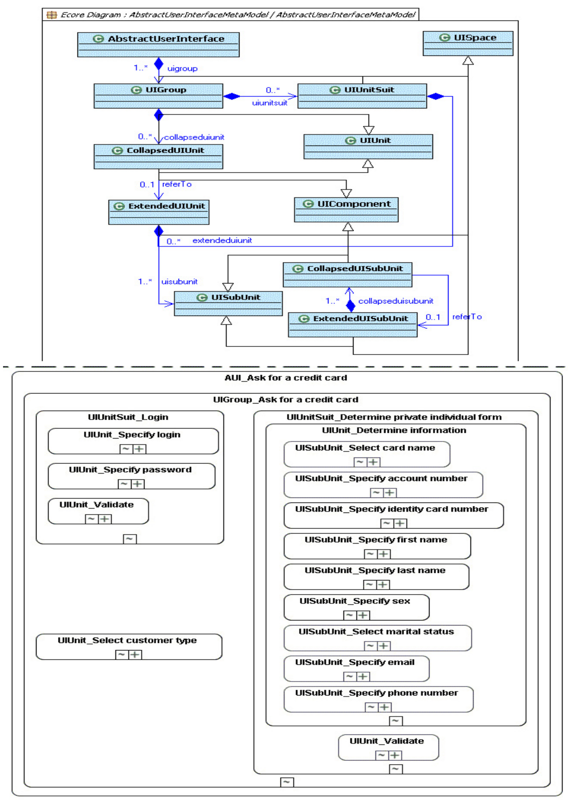


Fig. 6. AUI Meta-model and AUI for the possession process of the credit card (case of a private individual customer)

Concrete User Interface (Meta-) Model. The Concrete User Interface (CUI) is deduced from the Abstract User Interface (AUI) to describe the interface in terms of graphic containers, interactors and navigation objects. It is also expressed through the BPMN notation. The CUI meta-model extended from the static model of the interactions of [7] is presented in Fig. 7. It consists of one or several windows presented in the meta-model by the "UIWindow" class. The "UIPanel" class allows the modelling of the possible hierarchies of containers. The interactors presented by the "UIField" class of the concrete interface are classified according to their types in three groups: "UIFieldMultimedia", "UIFielData" and "UIFieldControl". The CUI presented in Fig. 7 shows a possible concretization of AUI (Fig. 6) for a small-screened platform.

Platform (Meta-) Model. Aiming at generating plastic interfaces, the platform meta-modelling has become a necessity in this work. Although most of the work on plastic UI made adaptation to the platform, the latter remains without a complete and detailed meta-model. The existing approaches only describe it at a high level of abstraction or describe only the display surface of the platform which represents the most used interactional resource in the adaptations made so far. However, the adaptation can be prepared in the presence and absence of the other interaction devices. For example, if we do not have a mouse, we can suggest as a form of adaptation using a vocal inter-actor where the activation of the actions will be made vocally.

Fig. 8 presents our platform meta-model. Generally, the platform consists of:

- Calculation resources represented in Fig. 8 by the "ComputationalCapacities" class. These resources include not only the material aspect, such as the memory or processor but also software aspect as the supported operating system;
- Interaction resources which are the input-output devices represented in our meta-model by the "InteractionDevices" class. We identify two classes of interaction devices: the input devices (InputDevice class in Fig. 8) and the output devices (OutputDevice class in Fig. 8). Certain devices inherit both classes and are thus input/output devices, such as the touch screen. As concrete example, in Fig. 8 we give also the tree-based description of "iPAQ HX2490 Pocket PC" realized by EMF-based editor.

Final User Interface (Meta-) Model. The Final User Interface (FUI) is operational i.e., the user interface works on a specific platform deploying a programming language. In our approach, the acquisition of the FUI is made by a "Model to Code" transformation type. In fact, we correspond to each CUI component a final representation using a particular tool box. Some work is in progress to produce HTML and Swing UIs. Fig. 9 presents some sketches for the "iPAQ HX2490 Pocket PC" platform of the PDA family.

4.2 Transformations Rules for Plasticity

The proposed approach of UIs development is defined by a series of models transformations, each of which takes input models and produces output models, until

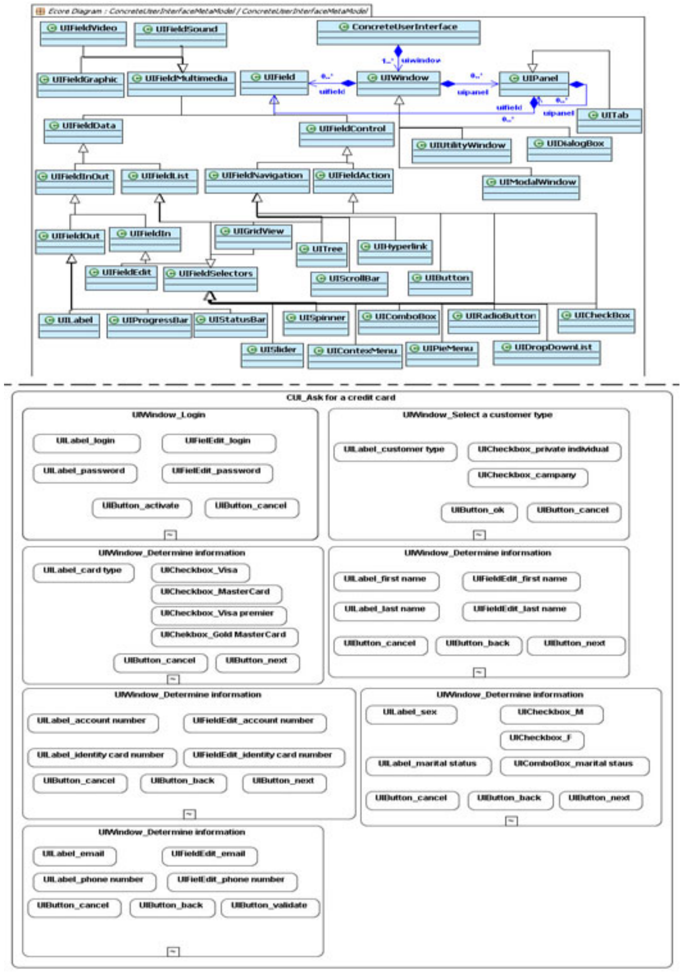


Fig. 7. CUI Meta-model and CUI for the possession process of the credit card (case of a private individual customer)

the obtaining of the final interface. This allows the HCI models (Concept Model, Task Model, AUI and CUI) to pass to the productive stage, which makes these models dependent on one another. The transformations are made explicit by the MDE; they are the core of the models engineering. Indeed, a transformation indicates a set of rules denoting the passage of a source model to a target model. For our approach, we used the Kermeta (Kernel meta-modelling) transformation language. Kermeta is created to allow the description of the meta-models as well as the definition of the constraints. It can also be considered as a language of action to give the operational semantics of the meta-models.

We propose two transformation modules. The first module is TMa2AUI having as source model the task model annotated by the concepts, allows the generation

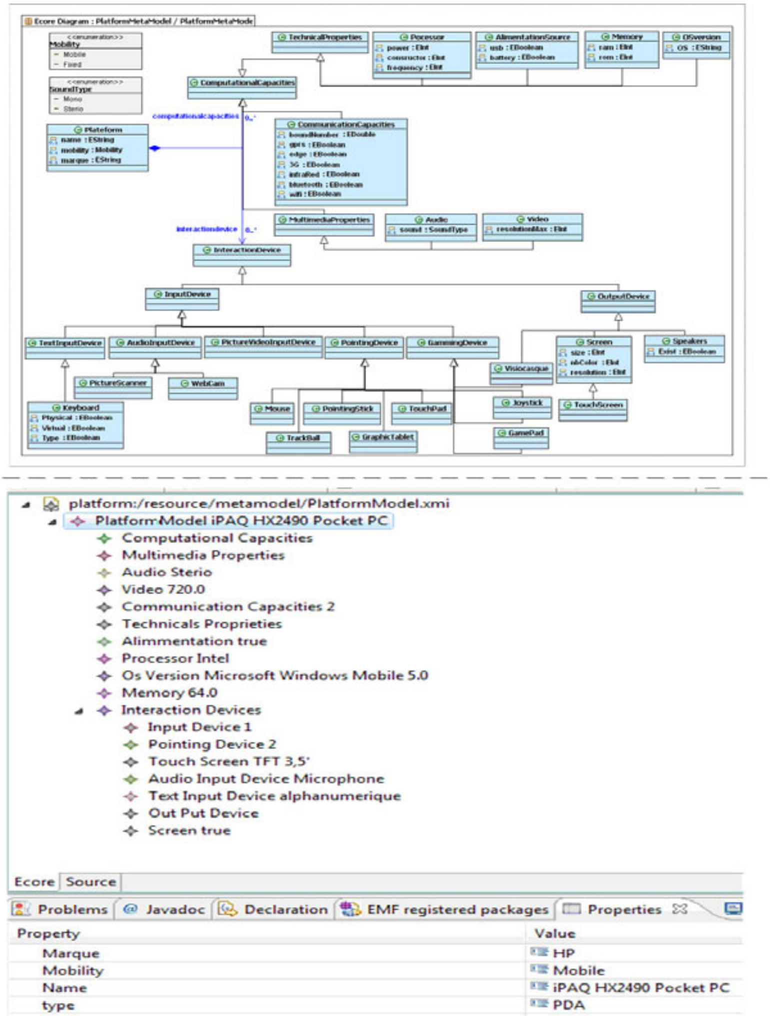


Fig. 8. Platform Meta-model and tree-based description of “iPAQ HX2490 Pocket PC”

of the AUI. Before introducing the task model into the transformation, it must be annotated by the domain concepts. This annotation allows the description of the matchings and the links between the concept model and the task model. In fact, the tasks treat the Concepts which are necessary for their implementation. In the approach of [27], the matchings between the models are done thanks to a mapping model which identifies several types of relations. In our approach, we exploited the artefact “Annotation” to attach to each elementary task a set of attributes and/or specified methods. For example, Fig. 10 shows the “Select card type” task annotated by the domain concept “cardType”.

The TMa2AUI transformation consists in creating an abstract user interface, their containers (UISpace) and abstract components (UIComponent) from the

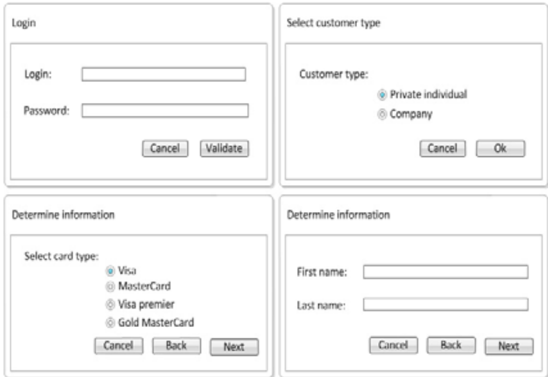


Fig. 9. Sketches for the possession process of the credit card (case of a private individual customer)

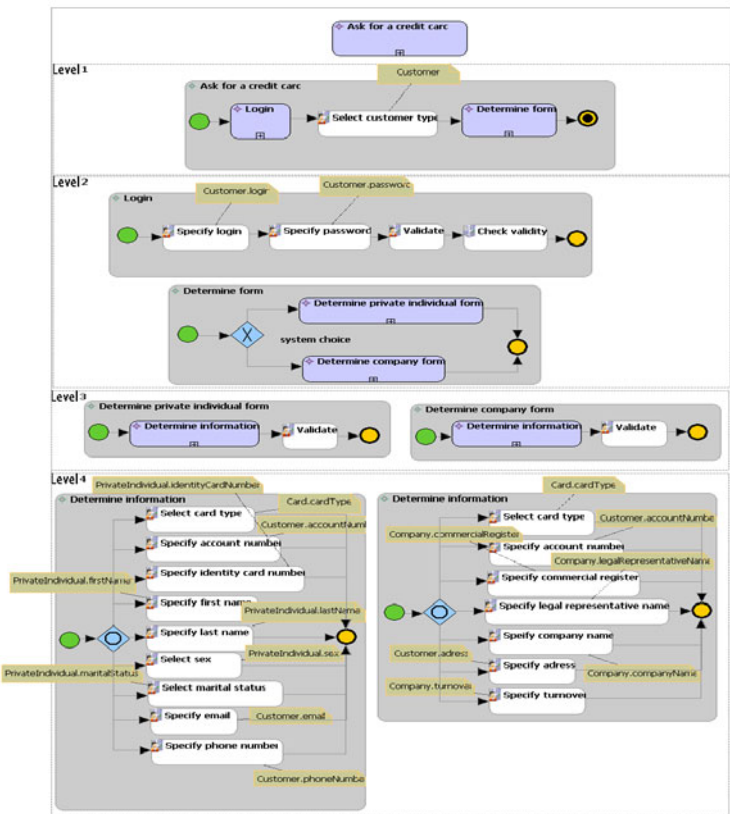


Fig. 10. Task Model "Ask for a credit card" annotated by the Concepts

annotated task model. In practice, this transformation can be implemented in Kermeta language by the following four stages:

– Creation of the application: creation of the application in the target model "AbstractUserInterface" by the "TaskModel" of the source model;
– Creation of the "UISpace";
– Creation of the "UIComponent";

To apply these stages, a set of rules is established. As an example, the extracted code below shows the instructions of "LevelsTreatment" method. It allows the creation of the "UIUnitSuit" container. Firstly, we have to recuperate the "CollapsedSubProcess" (CSP) from the "ExtendedSubProcess" (ESP) belonging to level one of the task model through the use of the "getCSPfromESP" method. Secondly, we have to create the "UIUnitSuit" instance in order to initialize and add them to the corresponding "UIGroup" (uig). Finally, we resort to the "ExtendContainerTreatment" method which is a reflexive method allowing the creation of the corresponding abstract containers of a container father.

```
operation LevelsTreatment(level : LevelOfDiagram, uig :UIGroup
,inputModel : TaskModel )
  is do
    getCSPfromESP(c).each{k|
    var uius : UIUnitSuit init UIUnitSuit.new
    uius.name := k.name
    uig.uiunitsuit.add(uius)
    ExtendContainerTreatment(level,uius,inputModel)
  end
```

The result of the transformation is an XMI file, which can be visualized by means of the AbstractUserInterfaceEditor developed with the Eclipse GMF plug-in which is based on the defined meta-model (Fig. 6). The first transformation allows having independent UIs of any modality of interaction and any implementation technology. However, the second module of transformation AUI2CUI which allows the generation of the CUI from the AUI is parameterized according to the target platform. This parameter setting presents the starting point of a standard description to a contextual description taking into account platform criteria. To do so, we build on the parameterized transformations defined by [34]. Vale [34] describes a parameterized transformation within the framework of the model driven engineering for a contextual development. The methodology proposed by [34] consists in defining the correspondences between the context model and the Platform Independent Model: PIM (Platform Independent Model) to define a CPIM (Contextual PIM). Then, an ordinary MDE transformation is used to define the CPSM (Contextual Platform Specific Model). The correspondences are assured by a parameter setting of the transformation. Its basic principle is to take into account the properties of the context during the transformation rules specification.

Based on the principle of contextualisation evoked by [34], we can use "the parameterized transformation" in the field of UI Engineering to consider the

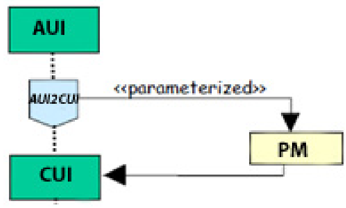


Fig. 11. AUI2CUI parameterized transformation

context of use. Such a transformation requires a triplet of models <Source, Target, Parameter>. The source model and the target model represent models of functional description (initial models) while the parameter model plays the role of a context model served for the contextualisation of the target model. Fig. 11 clarifies the transformation principle parameterized in our scenario. The parameter setting of the transformation leads to a strategy generation of UI adaptable to the context of use. We apply this parameter setting at the level of the AUI2CUI transformation. The context of use, taken into consideration, concerns particularly the platform.

The extract of the code below shows the operation "transform" which takes as a parameter two models: the input model (InputModel) which represents the AbstractUserInterface and the PlatformModel as adaptation model.

```
operation transform(inputModel :AbstractUserInterface
,adaptModel :PlatformModel) :ConcreteUserInteface is do
(The rest of code)
```

The generation stages of the Concrete User Interface lean strongly on the work of [32] and [20]. The AUI2CUI transformation can be implemented in Kermeta language by the following four stages:

– Creation of the application: creation of the application in the "ConcreteUserInterface" target model by the "AbstractUserInterface" of the source model;
– Realization of the abstract containers;
– Choice of the interactors;
– Definition of the navigation.

We have developed a set of rules allowing the transformation of an AUI into a CUI. As an illustration, in what follows, we clarify the stage of interactor's choice. This stage aims at associating the adequate interactor with the abstract component of AUI. Such a choice depends on the properties of the abstract component: its type (Input or Output) its nature (Specify, Select or Turn) and the platform properties (screen size, presence or absence of a keyboard, a mouse, a microphone etc. . .).

The UIField class of CUI meta-model presents a generalization of the various forms of interactors. The extract of the following code transforms every abstract component of the "CollapsedUIUnit" type into a "UIField" and appeals to the

"UIFieldTreatment" method for the choice of the appropriate interactor. In that case, it is a question of executing the interactor's choice for an abstract component of the "Specify" nature. We treat two cases as examples:

- If we have a keyboard or a touch screen, the "UIField" will be specialized in a "UIFieldIn" and a "UIFieldStatic".
- Else if we have input device of type "microphone" or "visiocasque", the "UIField" will be specialized in a "UIFieldSound".

```
operation TransformationTreatment(aui : AbstractUserInterface
,uiw : UIWindow ,p : PlatformModel)
  is do
    getAllCollapsedUnit(aui).each {cui|
    UIFieldTreatment(aui,p, cui,uiw)}
  end

// UIField specification
operation UIFieldTreatment( inputmodel : AbstractUserInterface,
paramModel : PlatformModel, cui : CollapsedUIUnit
, uiw : UIWindow)
  is do
  // recovery of annotation
  var lnk : Link
  lnk := getLinks(inputmodel)
  .detect{c|c.uicomponent.name== cui.name}
  var nat : Nature init cui.nature
  var tp : AnnotationType init lnk.uicomponentannotation.type

  if (MouseExist(paramModel) and ScreenExist(paramModel)
  and KeyboardExist(paramModel)) or (TouchPadExist (paramModel)
  and ScreenExist(paramModel) and KeyboardExist (paramModel))
  or TouchscreenExist(paramModel) then

  // Treatment of abstract component of type "Specify"
     if (nat == Nature.Specify) then
        createFieldIn(uiw,cui,lnk) // Creation of UIFieldIn
        createUILabel(uiw,cui,lnk) // Creation of UILabel
     end
rest of code
  else if VisiocasqueExist(paramModel)and MicroExist(paramModel)
  then
     createFieldSound(uiw,cui,lnk) // Creation of UIFieldSound
rest of code
  end
```

5 Conclusion and Perspectives

In this paper, we have presented a methodology for the development of the plastic UI of an Information System. Based on a BPMN workflow model, the processes are determined. Each process is detailed in a task model which follows a series of models transformations, according to an MDE approach to have the final UI. We proposed an extension in the BPMN notation to support the task modelling. To apply "model to model" transformations, we set up three meta-models: Task meta-model, Abstract User Interface meta-model and Concrete User Interface meta-model. The characteristic of the interface adaptation to its context of use was among our objectives. In order to reach them, we proposed a platform meta-model describing the material and software constituents of the interaction platform.

Our approach is distinguished from the existing approaches by:

– The use of a standard notation for the modelling of the majority of our approach models.
– According to the literature, the task modelling is, for the first time, made through the BPMN notation.
– The proposition of a complete and detailed meta-model for the platform. Encountered by new platforms, a definition of a model for this platform will be enough. So, our transformations rules are generic.

We foresee multiple perspectives for our work. These perspectives concern a meta-modelling of the environment and the user and the integration of the ergonomic properties in our transformations.

References

1. Allaguy, M., Trabelsi, W.: Crédit Scoring: L'octroi des cartes bancaires. In: Mémoire de fin d'étude, Institut des Hautes Etudes Commerciales de Carthage, Université de 7 Novembre á Carthage, Tunisie (2009)
2. Bézivin, J., Blay, M., Bouzeghoub, M., Estublier, J., Favre, J.-M.: Action spécifique CNRS sur l'Ingénierie Dirigée par les Modèles, Rapport de synthèse, janvier (2005)
3. Bézivin, J.: In Search of a Basic Principle for Model-Driven Engineering. Journal Novatica, Special Issus (2004)
4. Bodart, F., Hennebert, A., Lheureux, J., Provot, I., Sacré, B., Vanderdonckt, J.: Towards a systematic building of software architecture: The TRIDENT methodological guide. In: Proceedings of 1st Eurographics Workshop on Design, Specification, Verification of Interactive Systems DSV-IS 1995, Vienna. Springer, Heidelberg (1995)
5. BPMI, Business Process Modeling Notation version 1.0., http://www.bpmn.org
6. Brossard, A., Abed, M., Kolski, C.: Modélisation conceptuelle des IHM: Une approche globale s'appuyant sur les processus metier. Ingénierie des Systèmes d'Information - Revue des sciences et technologies de l'information, Numéro 2007/5, France, pp. 69–108 (2007)
7. Calvary, G., Coutaz, J., Thevenin, D., Limbourg, Q., Bouillon, L., Vanderdonckt, J.: A Unifying Reference Framework for Multi-Target User Interfaces. Interacting with Computers 15(3), 289–308 (2003)

8. Daassi, O.: Les Comets: une nouvelle génération d'Interacteurs pour la plasticité des interfaces Homme-Machine. Thèse de doctorat de l'université Joseph Fourier (2007)
9. Davenport, T.: Process Innovation: Reengineering work through information technology. Harvard Business School Press, Boston (1993)
10. Favre, J.-M., Estublier, J., Blay-Fornarino, M.: L'ingénierie dirigée par les modèles. Hermès-Lavoisier, Paris (2006)
11. Favre, J.-M.: Toward a Basic Theory to Model: Model Driven Engineering. In: Workshop on Software Model Engineering, Wisme 2004, Lisbonne, Portugal (2004)
12. Fingar, P., Bellini, J.: The Real-Time Enterprise. Meghan-Kiffer Press, New York (2004)
13. GMF, Graphical Modeling Framework, http://www.eclipse.org/gmf
14. Griffiths, T., Barclay, P.J., Paton, N.W., McKirdy, J., Kennedy, J.B., Gray, P.D., Cooper, R., Goble, C.A., Silva, P.P.: Teallach: A Model-based user interface development environment for object databases. Interacting with Computers 14(1), 31–68 (2001)
15. Guerrero, J., Lemaigre, C., Gonzalez, J.-M., Vonderdonckt, J.: Model-Driven Approach to Design User Interfaces for Workflow Information Systems. Journal of Universal Computer Science 14(19), 3160–3173 (2008)
16. Guerrero, J., Vanderdonckt, J., Gonzalez, J.M.: FlowiXML: a Step towards Designing Workflow Management Systems. Journal of Web Engineering 4(2), 163–182 (2008)
17. Kleppe, A., Warmer, J., Bast, J.: MDA Explained-The Model Driven Architecture: Practice and Promise. Addison-Wesley, Reading (2003)
18. Koch, N., Kraus, A., Cachero, C., Melié, S.: Integration of Business Processes in Web Application Models. Journal of Web Engineering (2004)
19. Kristiansen, R., Traetteberg, H.: Model-Based User Interface Design in the Context of Workflow Models. In: Winckler, M., Johnson, H., Palanque, P. (eds.) TAMODIA 2007. LNCS, vol. 4849, pp. 227–239. Springer, Heidelberg (2007)
20. Limbourg, Q., Vanderdonckt, J.: UsiXML: A User Interface Description Language Supporting Multiple Levels of Independence. In: Matera, M., Comai, S. (eds.) Engineering Advanced Web Applications, pp. 325–338. Rinton Press, Paramus (2004)
21. MDA, Model Driven Architecture, http://www.omg.org/mda
22. Mirbel, I.: Conciliating User Interface and Business Domain Analysis and Design OOIS 2003. Rivieres V.2003, pp. 383–399 (2003)
23. Mori, G., Paternò, F., Santoro, C.: Tool Support for Designing Nomadic Applications. In: Proceedings of the International Conference on Intelligent User Interfaces, Miami, pp. 141–148 (2003)
24. OMG, Object Management Group, http://www.omg.org
25. Salimifard, K., Wright, M.: Petri net-based Modelling of Workflow Systems: An Overview. European Journal of Operational Research 134, 664–676 (2001)
26. Sottet, J.S., Calvary, G., Favre, J.-M.: Ingénierie de l'Interaction Homme-Machine Dirigée par les modèles. In: Proceeding of IDM 2005, Paris, France, pp. 67–82 (2005)
27. Sottet, J.S., Calvary, G., Favre, J.-M.: Mapping Model: A First Step, to Ensure Usability for sustaining User Interface Plasticity. In: Proceedings of the MoDELS 2006 Workshop on Model Driven Development of Advanced User Interfaces (October 3, 2006)

28. Sousa, K., Mendonça, H., Vanderdonckt, J., Rogier, E., Vandermeulen, J.: User Interface Derivation from Business Processes: A Model-Driven Approach for Organizational Engineering. In: Proc. of 23rd ACM SAC 2008, pp. 553–560. ACM Press, New York (2008)
29. Sukaviriya, N., Kumaran, S., Nandi, P., Heath, T.: Integrate Model-driven UI with Business Transformations: Shifting Focus of Model-driven UI. In: Proc. of MD-DAUI 2005. CEUR Workshop Series, vol. 159 (2005)
30. Sukaviriya, N., Sinha, V., Ramachandra, T., Mani, S., Stolze, M.: User-centered Design & Business Process Modeling: Cross Road in Rapid Prototyping Tools. In: Baranauskas, C., Palanque, P., Abascal, J., Barbosa, S.D.J. (eds.) INTERACT 2007. LNCS, vol. 4662, pp. 165–178. Springer, Heidelberg (2007)
31. Terrase, M., Savonnet, M., Leclercq, E., Grison, T., Beker, G.: Points de vue croisées sur les notions de modèle et métamodèle. In: IDM 2005 Premières Journées sur l'Ingénierie Dirigée par les Modèles, Paris 30 juin, 1 juillet (2005)
32. Thevenin, D.: Adaptation en Interaction Homme-Machine: Le cas de la Plasticité. Thèse de doctorat, Université Joseph Fourier, Grenoble I, pp. 212 (2001)
33. Traetteberg, H.: Integrating Dialog Modeling and Domain Modeling - the Case of Diamodl and the Eclipse Modeling Framework. Journal of Universal Computer Science 14(19), 3265–3278 (2008)
34. Vale, S., Hammoudi, S.: Context-aware Model Driven Development by Parameterized Transformation. In: Proceedings of MDISIS (2008)
35. Vanderdonckt, J.: A MDA-Compliant Environment for Developing User Interfaces of Information Systems. In: Pastor, Ó., Falcão e Cunha, J. (eds.) CAiSE 2005. LNCS, vol. 3520, pp. 16–31. Springer, Heidelberg (2005)
36. Workflow Management Coalition, http://www.wfmc.org/

Mal-processes: Explicitly Modelling the Deviant

David Sundaram and Max Erik Rohde

Department of Information Systems and Operations Management, The University of Auckland, Private Bag 92019, Auckland 1142, New Zealand
{d.sundaram,m.rohde}@auckland.ac.nz

Abstract. Definitions of business processes for the most part suggest that they are a set of interrelated activities with the goal of accomplishing a given task to provide value to the customer of the process. This positive orientation has been reflected in business process modelling methodologies, tools, techniques and even in the reference models and implementation approaches suggested by vendors and implementation partners. Research and industry does not explicitly consider the negative business process scenarios that could result in the accomplishment of undesirable outcomes to the customer and/or owner of the process. In this paper we first motivate the need for explicit consideration of such undesirable processes. We define such processes as mal-processes and proceed to identify some of the key causes of mal-processes in organisational contexts. This discussion is motivated through the identification of potential mal-processes and means through which we could prevent them. We propose extensions to existing business process modelling conventions (ARIS) that would enable us to model mal-processes. The interplay between best practice processes as defined by reference models (e.g. SAP Reference Model) and mal-processes as well as their impact on the implementation of Enterprise Systems (e.g. SAP, Oracle) is discussed in some detail. We conclude the paper by identifying the application and benefits that could accrue from the explicit identification of mal-processes.

Keywords: business process modelling, undesirable processes, mal-processes, reference models, enterprise system implementation.

1 Introduction: Preventing Behaviour to Be Avoided

With the financial crisis, organisations are expected to be under greater risk to be subject to fraud [1]; a situation that might persist in an increasingly complex and fierce business environment [2]. Organisations employ a great number of systems to detect and prevent fraud [3]. However, this is just one example of how ‘things can go wrong’ in an organisational environment.

Other potential causes can for instance be human errors [4]. What both fraud and human error have in common is that their potential causes can be seen as twofold: originating from individual cases or mistakes or from systematic shortcomings in the system. These systematic shortcomings can for instance be that likely cases for failures are not considered in the system [4]. Significant research on high-reliability

J. Barjis (Ed.): EOMAS 2010, LNBIP 63, pp. 164–178, 2010.

organisations explores the ways in which such organisations, for instance atomic power plants, handle these risks. One key aspect is that these organisations try to anticipate very many different cases, which might occur. One way to accomplish this is, of course, by modelling these cases in a structured way.

Prior research in modelling has proposed mechanisms, in which organisations can model threats for the appropriate execution of routines, for instance in misuse cases [5, 6].

In our research, we want to explore how organisations can explicitly model the process dimension of threats of execution of routines. This goes along with a rising awareness of the security dimension of business processes [7, 8].

2 Model Management, Business Process Management and Risk Management

Model management traditionally aspires to capture parts of organisational reality in a structured life cycle [9]. Emphasis lies thereby in the accuracy of these models and the ability to base organisational decisions upon them. These models often have the ability to enable mathematical precision in organisational decision making. Models fall in different organisational domains, but further can potentially bridge and integrate these domains [10]. Traditional model management accommodates for model, which relate to both the wanted and unwanted. Mathematical models can be formulated, for instance, to predict critical conditions for nuclear power plants or create forecasts for bad market condition.

As our focus is on the process perspective of organisations, we are particularly concerned with models, which capture organisational processes or business processes. Different comprehensive modelling frameworks have been proposed, which account for the process perspective; among these are the Architecture of Integrated Information Systems (ARIS) [11] and the Semantic Object Model (SOM) [12]. Especially ARIS finds its application in the domain of business process management [13, 14]. However, most business process management frameworks follow an approach, which is driven by the best business practice, which is to be reflected in the design of processes and their implementation as well as monitoring. How things are *not* supposed to be, is usually not considered in business process management approaches.

According to Lyytinen, Mathiassen and Ropponen [15], in rational decision theory, risk can be defined as “the variation in the distribution of possible outcomes, their likelihoods, and their subjective values.”. Risk management is often associated with a particular domain, for instance software risk management [15, 16], risk management in banks [17], project risk management [18], and risk management for business processes [19, 20].

In risk management for business processes, it has been proposed that many aspects of traditional business process models can be associated with a certain risk. Rosemann and zur Muehlen [19], for instance, give the example that an activity “Enter payroll run information” can be associated with the risk of “data entry mistake”.

Best business processes are a very useful guideline for organizations to design processes. However, in the enactment of routines, these ideal processes are not always achieved. An important concept which has received less attention than best business processes is their opposite; the conceptual notion of how a business process should in no case be conducted. These have been conceptualised, for instance, as misuse cases, abuse cases, or failure cases [5, 6, 21, 22].

Strong arguments have been brought forward that both best business processes and their opposite shall be modelled to enable organisations to deal with these processes in a structured fashion. Model management systems are a capable concept to deal with many issues arising in the management of these models. Best business processes are a central component of business process management. Business process management architectures provide wide-ranging support for the management of business processes to help organisations strive towards the best business practices. However, these architectures often do not consider how things are not supposed to be.

There is conceptual support for both the modelling of best business practices as well as mal-processes. Best business practices further drive many business architectures and system. However, there is little attention on how an architecture or system could be designed, which accommodates for both the best way of managing processes and the major risks, which may occur in the course of a process execution.

Our review shows that attempts, which focus on supporting risk management and business process management in an integrated fashion, are sparse. Of course, process monitoring is a central aspect of business process management. But this monitoring primarily relates to expected outcomes rather than the unexpected. One reason for this is, that in order to be able to monitor a process with the means of business process management, the process must be known in all its varieties; because only then performance indicators for the individual elements of the process can be set up. In result, business process management is very strong, when it comes to processes, which are well-defined, known and work as expected. These processes are most likely healthy as they are well-understood and well-monitored. The unhealthy processes, those that potentially harm the organisation, however, are often unexpected and in total or in parts unknown to the organisation. Fig. 1 illustrates what we see as the process management chasm and have discussed in this section.

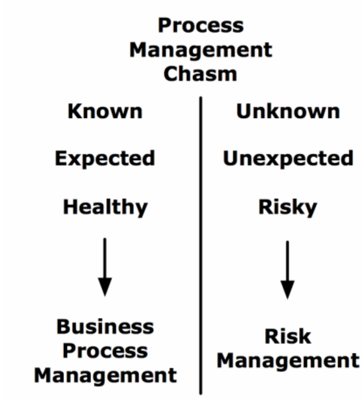


Fig. 1. The Process Management Chasm

In the following, we want to bring forward an extension to ARIS, which can help in the design and implementation of business process management systems, which consider both best-practices business processes and undesired processes.

3 Mal-processes

When executing a particular process, a user may intentionally enter incorrect information into the database, or modify the existing data in the wrong way: for example execute an incomplete business process, or execute a wrong branch of the business process, or even create an new branch of a business process that is undesirable. We argue, that these situations mostly could be avoided at the design stage, rather than having to deal with them as they occur.

We focus on the actions, which may be executed because of negligence or accomplished with intent. We do not consider similar effects that could be produced accidentally, as they are a subject of interest for data safety. The focus of this paper will be on preventing the possibility of creating or executing such undesirable processes during the design stages of the business process. We term such processes as *mal-processes*.

Mal: bad(ly), wrong(ly), improper(ly) (New Shorter Oxford English Dictionary, 1997)

Processes: A business process is a collection of interrelated tasks, initiated in response to an *event*, that achieves a specific *result* for the *customer* of the *process* [23]. Hereafter we refer to such processes as *regular processes* to distinguish them from *mal-processes*.

Mal-process can therefore be defined as a collection of interrelated tasks (executed in the place and time assigned for a *regular process*) that can result in harm for the customer or stakeholder of the process.

Hence, a mal-process could be considered as an undesirable branch of the regular/normal/complete business process, triggered by the same set of events, but achieving an undesirable result for the *customer* of the *process*. Therefore mal-processes are behaviour to be *avoided* by the system. Mal-processes are a sequence of actions that a system can perform, interacting with a legal user of the system, resulting in *harm* for the organization or stakeholder or customer of the process if the sequence is allowed to continue or complete.

Especially in the context of high-reliability organizations it is often attempted to be able to anticipate the unwanted in order to have measures in place to prevent it [24]. The organizations must "mind" what can happen in unwanted instances.

The consideration of mal-processes is extremely important from the security point of view. But apart from that it has significant consequences for efficiency. Current reference models and implementations of enterprise systems do not consider mal-processes explicitly. Addressing of mal-processes involve business oriented decisions that need to be considered by business analysts up front rather than by technical configuration experts later on during the implementation.

Routines are usually enacted differently in every instance [25]. Thereby it can be assumed that some of these enactments are 'better' or 'worse' than others. Reason for the occurrence of bad enactments may be based on individual or systematic error [4]. An organisation could, for instance, be too inflexible to react to a changing environment and a way of conducting a routine, which was a good practice before, turns into a bad practice [26].

Mal-processes are similar to *misuse cases, abuse cases, failure cases* [5, 6, 21, 22]. However, there is an important difference between them. *Misuse cases* and *abuse cases* assume hostile intent of an internal or external actor, so they are mostly concerned with the security of the system. In contrast mal-processes do not suggest an external hostile influence. Thus mal-processes are strictly not in the domain of systems and data security. Rather they are subject of study for those involved in the design of systems and those who seek efficiency and effectiveness of processes and systems.

However, the results of a mal-process are quite similar to the results of misuse or abuse cases: the systems may function in an unexpected and undesirable fashion. It incurs losses to the major stakeholders, or to the organisation as a whole but may benefit the person who is involved in the direct execution of the process. So, from the point of view of the organisation, a mal-process is simply a poorly/badly designed business process: the intent is correct, but the result may be different from that expected or hoped for.

Hence, we suggest that the field of mal-processes is a separate field of study. While it is close to the fields of data safety and data security, it does not intersect with them.

4 Causes of Mal-processes

Obviously there could be many causes to mal-processes. We identify below a representative sample of causes for mal-processes in a business context:

- Conflict of interests between an organisational unit (here we assume a representative of the unit who is a direct user of the system) and the whole organisation.
- Excessive workloads of the organisational units; as a result a part (or even the whole) of the business process being neglected or being executed in a careless fashion.
- Very deliberate violation of the normal process because of a material, pecuniary and/or immoral incentive/reasons.
- Organisation and/or individuals may adopt mal-processes without realising they are mal-processes (mal-processes by ignorance).

We discuss each of these in more detail with examples and means of addressing them in the following paragraphs.

4.1 Mal-process as a Result of a Conflict of Interests

All users have a position in an organisational structure. The structure is a hierarchy of positions, connected by the relation "superior-subordinate". We shall call the users independent, if there is no "superior-subordinate" chain between them in the organisational hierarchy. One of the reasons why mal-processes occur is a conflict of interests, when users have to enter information into the system that can harm their own position in a hierarchy, or their superior's position.

Conflict of interests may affect the normal course of a business process when for example; a part of the business process is employee reporting on her own performance.

In this case the employee might be inclined to:

- exaggerate his/her achievements,
- diminish their faults
- cover excessive waste in materials, money, labour or equipment use due to sub-standard quality of production and/or services.

In this situation the mal-process can be avoided by:

- changing the user of the process, so the reporting is done by another independent employee, preferably higher up in the hierarchy,
- duplicating the reporting process in another business process.

Examples

Customer Relationship Management

A company runs a Customer Relationship Management (CRM) system. The customer relations manager is a subordinate of the Executive Director. One of the functions of the system is to collect feedback from customers, process it and to report the company’s operational performance to the Board of Directors and to the shareholders. Because the Executive director (through the customer relations manager) has an interest to hide faults and to reveal only “good news”, it is better to transfer the reporting of the CR manager directly to the Board, or even better to transfer this function to an independent body. The CR manager can distort the process of collecting the feedback from customers so that only the favourable response will surface.

Production

A line manager after completing the daily assignment puts data about the actual performance (feedback) into the database. The basic business process is realised through SAP shop-floor control modules, COMPLETION OF ORDER in particular. Very frequently this data is incomplete or inaccurate or controversial, because it represents the results of line manager’s work. Sometimes, the results are poor because of errors in work organization that are direct faults of the line manager. In this situation this business process cannot be transferred to any other person (higher in the hierarchy), like the shop manager, because it will involve too much additional work for the shop manager, who is not involved in the operations management of the line. By duplicating the reporting process in another business process, say in the “Receipt of Finished Goods”, designed for the manager of the finished goods store, the validity of the information not only can be verified, but also significantly improved. When the line manager knows that her data will be soon double-checked, she will make much less mistakes. On the other hand, the process cannot be completely transferred to the manager of the store of finished goods, because the line manager puts the data into the database immediately after the order completion, while the goods reach the store of finished goods much later. The time lag sometimes might be unacceptable.

4.2 Mal-process as a Result of Excessive Workloads of the Organisational Units

Excessive workloads of the organisational units may affect the normal course of a business process, where the user simply does not have enough time or energy to execute the whole process.

In this case the user may:

- execute only a part of the process that she considers the most important,
- execute the process only in situations that she considers important,
- completely neglect all the procedures: “If I had time, I would find something more useful to do”.

Example

As a typical example of such procedures we can suggest ‘physical count’ that is a regular business process in inventory management of any ERP system, SAP included. The process is triggered by the calendar – this is a periodic procedure (monthly, quarterly). The goal of the procedure is to keep the inventory records in the database adequate to the physical levels. The record of the database for every item is compared with physically available stock. A standard for natural losses is set. If the difference is less than the standard, the situation is normal. If the difference exceeds the standard, the situation is recorded for the reconciliation committee, which will analyse the cause of shortage. In both situations the record is updated according to the physical count.

This procedure is a very important pillar of the database. Effectiveness of many business processes is pinned to the accuracy of inventory records: production planning, sales, accounting, etc. Inaccurate inventory records will cause the mal-functioning of practically all ERP systems.

The main problem with this process is that it requires too much physical effort from staff already loaded with other duties. Instead of climbing the ladders and counting nuts and bolts the store managers are inclined just to ‘tick the box’. Sometimes they count only the most important inventory of A and B class, and sometimes do not count at all.

This problem was identified long before the development of ERP systems; however traditional management depends not so dramatically on the accuracy of information. And when in doubt, the manager always could call and ask for a physical count of a particular item. Quite to the contrary: the super efficiency of ERP requires super accuracy in data, and the ‘speed-to-market’ quality of ERP cannot be compromised by double-checking delays.

So, the remedy is to identify the possibility of mal-functioning of a business process, to stop its use, and to design a correct and effective process. For example, the complete physical count by the staff of the store can be supplemented by a sample count carried out by an independent person.

4.3 Mal-processes Deliberately Caused by a User That Has a Material Pecuniary and/or Immoral Incentive/Reasons to Do This

This category includes, among other mal-processes, attempts to cover up petty theft of raw materials or finished products. Petty theft of raw materials or finished products frequently occurs in the food industry and in consumer goods production. If the thief is the user or his close associates, then he is inclined to cover up the theft by putting false information in the system. Apart from direct financial loss, it negatively affects the whole management system because of unreliable information about actual amount of raw materials and finished goods.

It seems that this negative action can be eliminated through physical counting. However, the physical count is conducted rather rarely, say once per month. Suppose that the theft occurred early in the month, and the physical count was carried out at the end of the month. The difference between the record and actual exceeds the standard; the situation is recorded for reconciliation committee, which will analyse the cause of shortage. By the time of analysis there will be no real possibility of discovering the guilty party, and most probably, the difference will be assigned to some input error, or to the inaccuracy of the bill of material.

The most useful best practice to cope with petty theft is a double-check by another process, assigned to an independent user.

Example

The store of raw materials provides a raw materials batch for the daily production, using the daily line schedule and the bill of material (BOM). The line has a small storage, in which raw materials not used during the day, are kept for the following day. At the end of the day the line manager sends the goods produced to the store of finished goods and puts the amount of produced goods into the database. The process assumes that some of the raw materials might not be used and remains in the line storage.

The excess of raw materials may occur for several reasons:

- The packed quantity of raw materials is not a perfect divisor of the required quantity. For example butter is packed in boxes by 25 kg each. Suppose, the required amount for the day is 30kg, then two boxes will be sent to the line; the remaining 20 kg are supposed to be used in next day's production.
- Due to the unplanned downtime the amount of finished goods produced may be less than planned, so not all of the raw materials were used.
- Sudden change of the daily schedule were authorised by top management. The new products require other materials, which were urgently requested and delivered to the line (in addition to the already delivered daily package, which will stay in the store of the line).

According to the design of the process, the line manager should report any unused (for any reason) quantities of materials to the store of raw materials. These unused quantities are supposed to be deducted from the next day's package. At the end of the month both stores make a physical count. This results in the creation of a list of materials sent to production. This list is compared with a similar list from the store of raw materials for reconciliation.

This business process is flawed, and gives a lot of space for petty theft of raw materials and finished goods in the factory. These lists may be significantly different, and there is no information that helps to find out the real reasons why. The obvious reason may be petty theft of the raw materials from the line storage, or theft of finished goods before they reach the store of finished goods. There may be other reasons, like an attempt of the line manager to conceal excessive waste. However, if something happened at the beginning of the month, and has been discovered at the end of the month, it is practically impossible to find out the real reason, given the practical absence of necessary information.

The correct organisation of this business process would entail a few significant changes. There must be an independent user that verifies the actual use of raw materials. Such an independent user might be the manager of the store of finished goods. The normative amount of raw materials used might be verified through the BOM. As soon as the store of finished goods receives the daily production, the store manager puts the amounts in the database. These amounts overwrite the amounts put by the line manager (not physically overwrite, but in any dispute this amount is accepted as correct). Before computing the daily pack of raw materials, the manager of raw materials store runs MRP on the actual amount produced the previous day. Thus she determines the actual amount of raw materials consumed the previous day and the actual amounts of raw material left in the line store. Any dispute involves the reconciliation committee which:

- has all the information about passing of the raw materials down the track.
- has a time lag between the occurrence of the loss and its discovery of no more than one day.

4.4 Mal-processes by Ignorance or Lack of Knowledge

Organisations and/or individuals may be quite often adopt mal-processes without realising that they are mal-processes. A typical example here may be the following. The enterprise system design and implementation group as usual consists both of the employees of the company and implementation consultants. The employees of the company as a rule promote and defend existing business processes. The reasons they give are:

- Their processes reflect the specific feature of company's functioning,
- If the employees saw the way to improve their processes, they would have certainly re-engineered them long before,
- The company uses these processes, and no harm has been detected; this is evidence of what may not be the best, but what are reasonably good business practices.

The real reason for opposing change is the employees of the company are used to their processes, and therefore frequently are not able to critically assess them. At the same time, these processes might produce a far from efficient management, and what is worse, might become unsuitable or even harmful in the ERP environment. However, the consultants of the implementation team may agree with the employees for political reasons (those who pay, are always right), or simply because of lack of knowledge.

Unfortunately, there is no good recommendations how to avoid such mal-processes, apart from the obvious: stick to the reference model as close as possible, with the hope that the ERP developing company really puts in the reference model the best business practices.

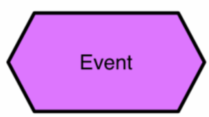
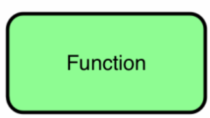
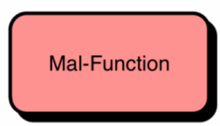
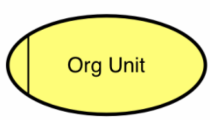
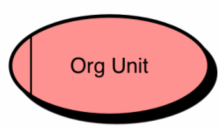
5 Modelling Mal-processes

Mal-processes can appear in an organization in many ways. First, they may appear as a result of poor design. In the reporting example above the designer may not see a reason for duplicating the feedback. As a result, the feedback process might be compromised. Second, even with the process correctly designed, one of the users may take a “shortcut”. In the same feedback example either the line manager or the manager of the finished goods store knows about the duplication of the feedback. Either one, or another (or even both of them) may decide not to put in the feedback data. Third, the process may be assigned to a wrong person, who will deliberately mutilate it, as in the example of CRM reporting.

All this requires that the mal-processes be recorded in the same place and in the same format as best practices business process.

Since mal-processes are equivalent in their characteristics to other processes except for their outcomes, we can use standardised process modelling syntax and constructs to represent mal-processes. To illustrate one example of how mal-processes can be modelled, we follow the syntax of event-driven process chains [27]. We will reuse three central elements of this modelling technique. (1) *Events* in event-driven process chains start and end every process. They represent occurrences in the environment, which trigger the process or are a result of the transition form one process step to the next. An example for an event could be “Minimum stock level reached”. (2) *Functions* represent activities, which need to be undertaken in order to transform inputs into desired outputs or change the environmental conditions as desired in the course of the process. For instance, “Order new stock” would be an example for a function. (3) *Organizational units* can be associated to functions and express which organizational entity is responsible for execution of the function. “Purchasing department”, for instance, would be an example for an organization unit, which could be associated to above given example for a function.

Table 1. Modelling Mal-Processes

Modelling Construct	Normal Process	Mal-process
Event		
Function		
Organisation		

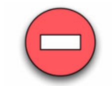


Fig. 2. 'No entry' sign

But the question arises regarding how we distinguish mal-processes from regular processes. One simple means of distinguishing is by colour but that may mislead if someone looks at a black and a white printout. Another option is to distinguish the events and functions of mal-processes from normal processes. Table 1 illustrates some of the means (colour, weight of line, and/or shadow) by which we could distinguish mal-processes from normal processes in the context of constructs used to build event-driven process chains.

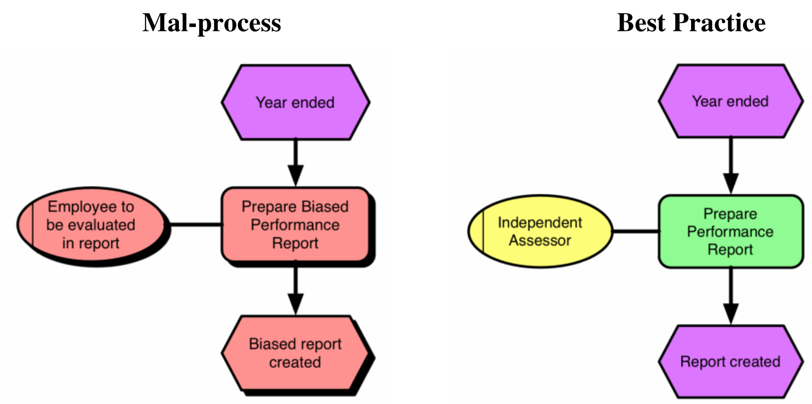


Fig. 3. Application of *Mal-process* and *Best Practice* or *Regular Process* Modelling Constructs

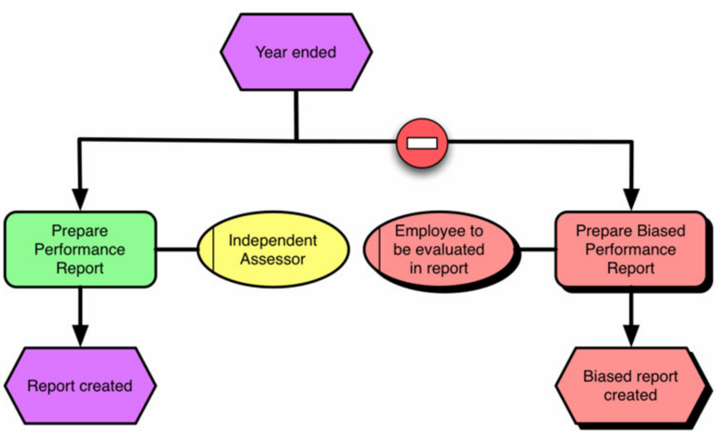


Fig. 4. A Model that integrates the mal-process and the regular process

If we model a mal-process as a branch of business process, we need a special operator as illustrated in Fig. 2 for:

- preventing the user to execute this branch of the business process and
- preventing the designer to design a mal-process

The application of these modelling constructs in the context of our first mal-process example is illustrated in Fig. 3 and Fig. 4.

6 Interplay between Best Business Practices and Mal-processes

Just as we have best business practices, mal-processes illustrate wrong and harmful business practices. So, it would be natural, by analogy with the reference model (that represents a repository of best business practices), to create a repository of wrong and harmful business practices. Thus a designer could avoid wrong design solutions and ultimately wrong workflows and incorrect enterprise system implementations.

There is though a very important difference between best and wrong business practices. While the best practice is unique, the ways in which one could go wrong are very many. Significant deviations from the best practice are wrong, causing harm to the effectiveness and efficiency of the management system. Thus it seems enough to put a warning sign on the reference model: *Do not deviate from the prescribed process!*

It is general knowledge that the reference model represents only a typical management system, and in this capacity it does not reflect the specific features of any particular enterprise. At the same time it is the specific features that are mostly responsible for the efficiency and effectiveness of the particular management system. The role of the designer is to create an enterprise system that keeps the integrity and efficiency of the reference model, reflecting at the same time all necessary specifics of the enterprise. Thus the designers deviate from the reference model as a rule, and not as an exception. Do they create mal-processes? No, not all deviations from the best processes are mal-processes.

We argue that there are some *stable changes* in the business processes, that we call mal-processes. The reasons they exist in the management system are explained in the section *"Causes of Mal-processes"* They are typical in the sense that they are independent fn the industry or the size of the enterprise. So, a mal-process is an intentional stable deviation from the best process, and its stability is explained by the intent of the user.

A detective hunting a thief has only one path she can follow, the path taken by the thief. But the thief can take any path she wishes to escape the detective. In a similar fashion, while the best practice in a particular situation maybe *one*, the mal-processes that could occur in that situation maybe *many*! Practitioners and researchers in the area of use cases face very similar problems. The ideal use case path usually known as the *basic flow* or *happy path* is *one*, while the alternative flows, *worst-case scenarios*, *variants*, *exceptions*, etc are *many*.

A repository of such mal-processes would enable organisations to avoid typical mistakes in enterprise system design. Apart from design such a repository could be used in a dynamic fashion to ascertain when a best practice process is deviating into a mal-process. Such a repository can be an invaluable asset in the education of ERP designers and it can be useful in general management education as well.

Another application of this repository might be troubleshooting. For example, sources of some nasty errors in the sales and distribution system might be found in the mal-processes of data entry in finished goods.

7 Conclusion

Applications and the ensuing benefits of this work are significant. Some of the key applications of our work are the incorporation of mal-processes as integral parts of:

- Modelling methods and tools such as ARIS
- Reference models such as SAP. Such integration would significantly impact on the Total Cost of Ownership of Enterprise Systems over a period of time. This could also shorten the time for implementation since some non-viable paths (mal-processes) are known beforehand. Not only would there be benefits at the time of implementation, we see ongoing benefits where the repository of mal-processes becomes an ever-growing repository that enables the organisations to keep to the straight and narrow.
- Enterprise Systems Analysis and Design methodologies such as Value SAP
- Implementation toolsets and/or accelerators such as SAP's *Implementation Assistant*, *Question and Answer Database*, *Diagram Explorer*, and *Business Engineering Workbench*. Integrating the concept of mal-processes into such tools would also impact positively on the enterprise system implementation process.
- Workflow Management Systems such that these mal-processes never get triggered or if they do get triggered they are immediately caught by the process intelligence/monitoring system and brought to the attention of management.
- Process management software of organisation to help them identify existing mal-processes as well as potential mal-processes.

Benefits of explicit incorporation of mal-processes into analysis, design, and implementation are many; we list just a few below:

- Early identification and resolution of mal-processes by incorporating it as part of the Enterprise Systems Analysis and Design lifecycle
- Potential savings to organisation and other stakeholders
- Identification of best business practices to remedy or prevent mal-processes would also be one of the benefits.

So, we suggest that a repository of typical mal-processes be created. Probably, the best place for such a repository is the reference models of vendors of enterprise systems such as the SAP Reference Model. The presence of models of mal-processes side by side with models of normal/best business processes would hopefully prevent the users of the reference model, be they analysts or consultants or process designers, from modifying/configuring the normal process into a mal-process. There are technical as well as organizational challenges for the creation and maintenance of such repositories. Obviously creation of such a repository and integration of the same as an integral of the organizational systems would need to be supported by the vendors of enterprises systems, workflow management systems, and business process management systems. The mal-processes to populate the repositories could be an exercise that could be undertaken by implementation partners, vendors, academics and ultimately the process modelers within organisations.

Acknowledgments. The authors would like to acknowledge the input of the late Dr. Victor Portougal into the early versions of this paper.

References

1. DeMarco, E.: Fraud and Misconduct Expected to Rise. RMA Journal 92 (2009)
2. Dale, S.: Holistic BPM: From Theory to Reality. In: Alonso, G., Dadam, P., Rosemann, M. (eds.) BPM 2007. LNCS, vol. 4714. Springer, Heidelberg (2007)
3. Albrecht, S., Albrecht, C., Albrecht, C.: Current Trends in Fraud and its Detection. Information Security Journal: A Global Perspective 17, 2–12 (2008)
4. Reason, J.: Human Error: Models and Management. BMJ: British Medical Journal 320, 768–770 (2000)
5. Sindre, G., Opdahl, A.: Eliciting security requirements with misuse cases. Requirements Engineering 10, 34–44 (2005)
6. Sindre, G.: Mal-Activity Diagrams for Capturing Attacks on Business Processes. In: Sawyer, P., Paech, B., Heymans, P. (eds.) REFSQ 2007. LNCS, vol. 4542, pp. 355–366. Springer, Heidelberg (2007)
7. Neubauer, T., Pehn, M.: Workshop-Based Risk Assessment for the Definition of Secure Business Processes. In: Second International Conference on Information, Process, and Knowledge Management, 2010. eKNOW 2010 (2010)
8. Backes, M., Pfitzmann, B., Waidner, M.: Security in Business Process Engineering. In: van der Aalst, W.M.P., ter Hofstede, A.H.M., Weske, M. (eds.) BPM 2003. LNCS, vol. 2678, pp. 168–183. Springer, Heidelberg (2003)
9. Krishnan, R., Chari, K.: Model Management: Survey, Future Research Directions and a Bibliography. The Interactive Transactions of OR/MS 3 (2000)
10. Wright, G., Chaturvedi, A., Mookerjee, R., Garrod, S.: Integrated Modeling Environments in Organizations: An Empirical Study. Info. Sys. Research 9, 64–84 (1998)
11. Scheer, A.: Business Process Engineering: Reference Models for Industrial Enterprises. Springer, Telos (1998)
12. Ferstl, O., Sinz, E.: Modeling of Business Systems Using SOM. In: Handbook on Architectures of Information Systems, pp. 347–367 (1998)
13. Scheer, A.-W., Nüttgens, M.: ARIS Architecture and Reference Models for Business Process Management. In: Business Process Management, pp. 301–304 (2000)
14. Heidasch, R.: Get ready for the next generation of SAP business applications based on the Enterprise Service-Oriented Architecture (Enterprise SOA). SAP Professional Journal (2007)
15. Lyytinen, K., Mathiassen, L., Ropponen, J.: Attention Shaping and Software Risk-A Categorical Analysis of Four Classical Risk Management Approaches. Info. Sys. Research 9, 233–255 (1998)
16. Boehm, B., DeMarco, T.: Software Risk Management. IEEE Software 14, 17–19 (1997)
17. Bessis, J.: Risk Management in Banking. Wiley, Chichester

18. Chapman, C., Ward, S.: Project Risk Management: Processes, Techniques and Insights. Wiley, Chichester (2003)
19. Rosemann, M., zur Muehlen, M.: Integrating Risks in Business Process Models. In: ACIS 2005 Proceedings (2005)
20. Lambert, J., Jennings, R., Joshi, N.: Integration of risk identification with business process models. Syst. Eng. 9, 187–198 (2006)
21. Alexander, I.: Modelling the interplay of conflicting goals with use and misuse cases. In: Proceedings of the HCI*2002 Workshop on Goal-Oriented Business Process Modeling. Memory of Dr. Maxim Khomyakov, GBMP 2002 (2002)
22. McDermott, J., Fox, C.: Using Abuse Case Models for Security Requirements Analysis. In: 15th Annual Computer Security Applications Conference, pp. 55–66. IEEE Computer Society, Los Alamitos (1999)
23. Sharp, A., McDermott, P.: Workflow Modeling: Tools for Process Improvement and Application Development. Artech House Publishers, Norwood (2001)
24. Weick, K.E., Sutcliffe, K.M., Obstfeld, D.: Organizing for high reliability: Processes of collective mindfulness. Research in Organizational Behavior 21, 23–81 (1999)
25. Feldman, M., Pentland, B.: Reconceptualizing Organizational Routines as a Source of Flexibility and Change. Administrative Science Quarterly 48, 94–118 (2003)
26. Orlikowski, W.: Knowing in Practice: Enacting a Collective Capability in Distributed Organizing. Organization Science 13, 249–273 (2002)
27. Davis, R.: Business Process Modelling with ARIS: A Practical Guide. Springer, Heidelberg (2001)

Modeling Human Decision Behaviors for Accurate Prediction of Project Schedule Duration

Sanja Lazarova-Molnar and Rabeb Mizouni

Faculty of Information Technology, United Arab Emirates University,
PO Box 17551, Al Ain, United Arab Emirates
{sanja,mizouni}@uaeu.ac.ae

Abstract. Simulation techniques have been widely applied in many disciplines to predict duration and cost of projects. However, as projects grew in size, they also grew in complexity making effective project planning a challenging task. Despite several attempts to achieve accurate predictions, simulation models in use are still considered to be oversimplified. They often fail to cope with uncertainty due to the complex modeling of the high number of interrelated factors. In this paper we propose a simulation model to cope with human resources uncertainty. We use the proxel-based simulation method to analyze and predict duration of project schedules exhibiting high uncertainty and typical human resources reallocation. The proxel-based simulation is an approximate simulation method that is proven to be more precise than discrete-event simulation. To model uncertainty, we introduce a new type of task, state-dependent (floating) task that supports and demonstrates a high degree of uncertainty in human resources allocation. In fact, it allows attributing different probability distributions to the same activity, depending on the team that may perform it. We use software development scheduling to illustrate our approach.

Keywords: Project scheduling, simulation, uncertainty, human resource allocation, on-the-fly decisions.

1 Introduction

Project planning is a discipline that is receiving a constantly growing attention. It is mainly because it has been identified as one of the success/failure factors of a project.

As projects grow in size, they also grow in complexity making effective project planning a hard task. It is mainly due to several interrelated factors that all have to be taken into consideration to predict in an accurate and precise way the cost and the duration of a project [1].

Many attempts have been conducted to improve the project scheduling problem [2-6], and each of them endeavors to offer an optimized schedule for a given project. Simulation is one of the techniques that has been successfully applied to project planning, e.g. in construction [7], where they use combined discrete-event/continuous simulation.

Duration of tasks in many project schedules cannot be modeled as deterministic [8], i.e. using fixed numbers. Consequently, probability distribution functions need to

J. Barjis (Ed.): EOMAS 2010, LNBIP 63, pp. 179–195, 2010.

be utilized to describe durations of tasks. Even with this assumption, the obtained simulation models are still considered limited and nonrealistic. They fail to take into account the different interrelated factors and uncertainty that in practice lead to plan changes. As stated by Joslin and Poole, “the simulation will be unrealistic if the plan is static” [1].

Human allocation uncertainty is seen as an example of a series of factors that can lead to a plan change. During the project run, based on the specific situations a team could be assigned a task that was originally assigned to another team if the latter one is unavailable (and the former team is available). The distribution function used to reflect the duration of the task most probably will be different to match the properties of the new task executers. Usually, teams have different levels of expertise which will in turn affect durations of tasks they perform. A practical simulation model should handle this dynamic aspect of a plan and anticipate possible changes.

In this paper we move one step towards the more realistic modeling of project schedules by incorporating a degree of anticipation of team allocation variability during the project execution. This uncertainty is modeled by means of human resource availabilities that are subject to unexpected changes.

We allow both duration and sequence of tasks to be variable – based on available resources and depending on various on-the-fly decisions by participants on the project. We simulate project schedules both with and without the possible on-the-fly decision scenarios. The objective of this is to study the effects of such changes in the project schedule and show their significance. Furthermore, we aim to provide an approach for accurate prediction of project schedule duration given the afore-described circumstances.

We use Gantt charts and state-transition diagrams for modeling project schedules. We extend both formalisms to model what we term as state-dependent (floating) task. Floating task represents a task that models uncertainty in human resources allocation and is a subject to various changes. We chose the proxel-based method for simulating the project schedules, as it has been successfully applied to this area [9, 10] and can provide highly accurate complete results.

Input probability distribution functions can be fitted based on historical data for similar tasks and situations and may be adapted to concrete situations of projects. The estimation process would, obviously, require a high level of expertise.

The remainder of this paper is structured as follows: In the next section we provide an overview of problems faced in the project scheduling simulation and we describe the proxel-based method. Further, we introduce the floating task model and our running example as a software development process. Then, we illustrate our simulation scenarios as well as our simulation results followed by a discussion on the importance of uncertainty and on-the-fly decision modeling in simulation model. Finally, we conclude.

2 Problem Definition

A project consists of a number of tasks (activities) where a predefined set of tasks has to be processed in order to complete the project. The tasks are in fact related by two constraints:

1. Precedence constraints: usually in a project development tasks cannot be undertaken in any order and some tasks cannot start unless others have been already completed; and
2. Resource sharing: performing tasks requires efficient resources' management. Such resources may include financial resources, inventory, human skills, production resources, information technology (IT), etc.

The incorporation of *uncertainty* into project planning and scheduling has resulted in numerous research efforts, particularly focusing on uncertainty in task duration or cost [4]. In our case, we are interested in studying the effect of human uncertainty factor on the duration of a project, in terms of on-the-fly decisions and resource allocation.

As running example, we consider a software development project. Typical requirements descriptions might include the task lists for people, and allocation schedules for resources.

Workforce allocation is seen as an important step in any software project management. It is the phase where all relevant elements of the software development process are taken into consideration for allocating software developers to the different project tasks [11].

While an initial allocation of software developers is calculated based on initial requirements, it is frequent that workforce adjustments during project performance becomes necessary for several reasons : (1) projections recalculation, based on the current workforce size, and the current development productivity [11], (2) number of remaining requirements to be implemented, and (3) requirements volatility.

Because of the above-mentioned reasons, it may happen that a team is assigned to a task that was originally assigned to another team during the workforce adjustment. Such scenario is in fact not easy to consider during the project scheduling. First, it is difficult to know when such adjustment will happen. This decision will be taken on-the-fly. Second, and more importantly, different teams have distinct expertise. Put in other words, the time that would take team A and team B to finish a task is not necessarily the same. It is frequent that one of the teams may need additional time to acquire the necessary expertise to achieve that particular task. When predicting the duration of project schedules, such scenarios should be considered.

The objective of our approach is to compute the probability distribution function of the duration of the project schedule taking into consideration human resource allocation uncertainty and typical on-the-fly decision behaviors. In addition we observe the effect that these behaviors might have on the duration on the project and want to stress the importance of their consideration.

3 The Proxel-Based Method

The proxel-based method [12, 13] is a relatively novel simulation method, whose underlying stochastic process is a discrete-time Markov chain [14] and implements the method of supplementary variables [15]. The method, however, is not limited to Markovian models. On the opposite, it allows for a general class of stochastic models to be analyzed regardless of the involved probability distribution functions. In other words, the proxel-based method combines the accuracy of numerical methods with the modeling power of discrete-event simulation.

The proxel-based method is based on expanding the definition of a state by including additional parameters which trace the relevant quantities in one model through a previously chosen time step. Typically this includes, but is not limited to, age intensities of the relevant transitions. The expansion implies that all parameters pertinent for calculating probabilities for future development of a model are identified and included in the state definition of the model.

Proxels (stands for probability elements), as basic computational units of the algorithm, follow dynamically all possible expansions of one model. The state-space of the model is built on-the-fly, as illustrated in Figure 1, by observing every possible transiting state and assigning a probability value to it (*Pr* in the figure stands for the probability value of the proxel). Basically, the state space is built by observing all possible options of what can happen at the next time step. The first option is for the model to transit to another discrete state in the next time step, according to the associated transitions. The second option is that the model stays in the same discrete state, which results in a new proxel too. Zero-probability states are not stored and, as a result, no further investigated. This implies that only the truly reachable (i.e. tangible) states of the model are stored and consequently expanded. At the end of a proxel-based simulation run, a transient solution is obtained which outlines the probability of every state at every point in time, as discretized through the chosen size of the time step. It is important to notice that one source of error of the proxel-based method comes from the assumption that the model makes at most one state change within one time step. This error is elaborated in [13].

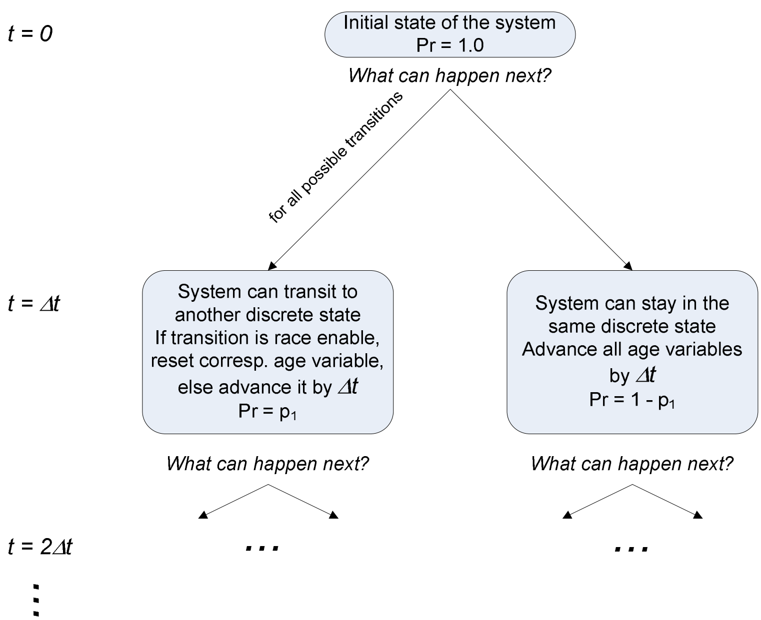


Fig. 1. Illustration of the development of the proxel-based simulation algorithm

Each proxel carries the probability of the state that it describes. Probabilities are calculated using the instantaneous rate function (IRF), also known as hazard rate function. The IRF approximates the probability that an event will happen within a predetermined elementary time step, given that it has been pending for a certain amount of time τ (indicated as 'age intensity'). It is calculated from the probability density function (f) and the cumulative distribution function (F) using the following formula:

$$\mu(\tau) = \frac{f(\tau)}{1 - F(\tau)} \tag{1}$$

As all state-space based methods, this method also suffers from the state-space explosion problem [16], but it can be predicted and controlled by calculating the lifetimes of discrete states in the model. In addition, its efficiency and accuracy can be further improved by employing discrete phases and extrapolation of solutions [17]. More on the proxel-based method can be found in [13].

4 State-Dependent (Floating) Task

4.1 Vital vs. Non-vital Tasks

To formalize uncertainty we define a highly uncertain *state-dependent task*, for which we allow any relevant parameters (including history of the project) determine its duration. We term this type of task as *floating* task. Its duration probability distribution is a complex function that among other factors depends also on the team that performs the task (its previous training, number of participants, etc.). The floating task supports introducing human decision uncertainty factors in project scheduling.

We classify all tasks into two categories, i.e. vital and non-vital, depending on their importance for the success of the project and the risk strategy of the project, i.e.:

1) Vital tasks: these tasks are estimated as critical for the success of the project. They are assigned only to *experienced professionals* to reduce the risk of their failure. Consequently, vital tasks are assigned a single team responsible for their implementation (fixed resource allocation strategy).
2) Non-vital tasks: these tasks are estimated as secondary for the success of the project. Non-vital tasks can be assigned to more than one team. Any of the teams that become available can implement it in order to optimize the project duration and maximize resource utilization. Under certain circumstances, non-vital tasks can be cancelled as well. In general, non-vital tasks invite various on-the-fly decision scenarios and can be often modeled by *floating tasks*.

Whether a task is vital or non-vital can be determined from the project requirements.

If we consider our running example, it is well known in software requirements management that users and stakeholders establish priorities to the feature set. Typical priority levels are: *critical*, *important*, and *useful* [18]. When simulating project schedules, we propose to categorize the set of prioritized features as vital and non-vital tasks. It is obvious that a critical feature with high risk cannot be seen as non-vital task since non-vital task may be even cancelled while a useful feature can be seen as a non-vital task. Introducing feature risk level as factor to decide about the categorization of the different features into vital and not vital tasks is out of the scope of this paper. Such issues are seen as part of our future work.

4.2 Case Study: The HOme Lightening Automation System (HOLIS)

The HOme Lightening automation System is a product to be marketed by Lumenations, a worldwide supplier of commercial lightening systems for use in professional theater and amateur stage production. HOLIS is a home lightening automation system that brings new lightening automation functionality with ease of use, comfort, and safety. It is intended to be used by homeowners' buildings and high-end homes. Details of the case study can be found in [18]. To simulate the schedule of the development of HOLIS, we select a subset of the different features that the system should implement. Table 1 summarizes the subset of HOLIS system features with their respective priorities and the effort needed to implement each of them. Effort has one of the three typical levels: *low*, *medium*, and *high*. Unsurprisingly, the effort needed to implement a task is team dependent. The values given in Table 1 represent an estimation of the effort needed to a trained team to implement the feature.

Table 1. Features of the HOLIS System

Features	Priority	Effort	Vital vs. NonVital
Feature 1: automatic timing settings for lights and so on.	Critical	Low	Vital
Feature 2: built-in security features (alarm, bells)	Critical	Medium	Vital
Feature 3: non-PC control unit	Critical	High	Vital
Feature 4: vacation settings	Important	Low	Vital
Feature 5: uses my own PC for programming	Important	High	Vital
Feature 6: close garage doors	Important	Low	Vital
Feature 7: automatically turn on closed lights when door opened	Useful	Low	Non-vital
Feature 8: interface to audio/video system	Useful	Medium	Non-vital

To simulate the project duration, we first need to determine vital and non-vital features. As shown in the last column of the table, the first six features are all considered vital. It is mainly because of their priorities (*Critical* and *Important*). The last two features are considered non-vital because they are useful features that the customers would like to have, only if possible.

4.3 HOLIS System: Teams Allocation

In order to simulate the HOLIS system, we determine the human resource allocation. We suppose that we have two teams working on the HOLIS project: Team A and team B. It is obvious that these two teams have different expertise, and each will fit better to a particular task. As explained previously, any software project is subject to workforce adjustment. During this case study, we anticipate such on-the-fly decisions and allow the expression of such possible decisions, while simulating the project duration. Table 2 illustrates the distribution of the features between the two available teams. It also shows the estimated effort that the assigned team needs to implement the features. All the vital tasks are assigned to a team and this team is not subject to change. However, the two non-vital features may be implemented by any of the available teams (A or B). The effort needed for these two teams to complete the task is not necessarily the same. As we can see, team A is more trained to implement Feature 7 than team B, while team B is more trained to implement Feature 8 than team A.

Table 2. HOLIS System: Teams Allocation

Features	Effort (for trained team)	Vital vs. NonVital	Assigned Team and Estimation of the Needed Effort
Feature 1	Low	Vital	Team A (Effort = *Low*)
Feature 2	Medium	Vital	Team A (Effort = *Medium*)
Feature 3	High	Vital	Team B (Effort = *High*)
Feature 4	Low	Vital	Team B (Effort = *Low*)
Feature 5	High	Vital	Team A and Team B (Effort = *High*)
Feature 6	Low	Vital	Team A and Team B (Effort = *Low*)
Feature 7	Low	Non vital	Team A (Effort = *Low*) or Team B (Effort = *Medium*)
Feature 8	Medium	Non vital	Team A (Effort = *High*) or Team B (Effort= *Medium*)

The conceptual model of the schedule is as shown in Fig. 2. This model clearly states that Feature 7 and Feature 8 will be implemented by any of the available teams, to the contrary of the remaining HOLIS system features. Thus, features 7 and 8 will be modeled using floating tasks.

To avoid risk, the vital features are scheduled first (in parallel), after what the teams spend a certain time on the non-vital features (7 and 8). If that exceeds a certain amount of time and takes too long, then both teams are transferred to features 5 and 6 that require combined work of both teams.

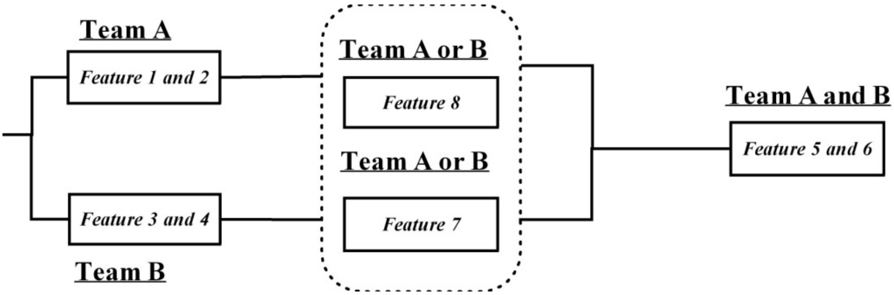


Fig. 2. Features Development Management

4.4 Simulation Model: Floating Task

In the following we present our sample model that we use to demonstrate our approach. The Gantt chart of the sample software development project schedule is shown in Fig. 3. Each of the tasks corresponds to a software requirement or combination of software requirements.

For simplicity reasons, we combine the features to represent tasks as follows:

- Features 1 and 2 into Task T1
- Features 3 and 4 into Task T2
- Feature 7 into Task T3
- Feature 8 into Task T4
- Features 5 and 6 into Task T5

thus resulting into a model with 5 tasks. Both, tasks T3 and T4 are floating tasks, and their processing depends on the state of the system including age intensities of tasks in progress.

The project schedule has two software developer teams assigned (A and B) and commences by running two tasks (T1 and T2, both vital) in parallel. Once either of the tasks is finished, the available team commences the task that is its favorable, i.e. lower effort (e.g. T3 for Team A, and T4 for Team B). When the team completes the task (T3 or T4), if the other team is still not finished on T1 or T2, then the former team starts working on the task that is rated as higher effort for it. If the originally assigned team completes finally its first task, and the other team has worked for a *short time* on the High Effort task, then it is cancelled, and both teams start working on Task T5. Alternatively, the available team waits for the other team to complete its task before both commence with T5.

Additionally, if the teams on the project are currently processing only non-vital tasks and it takes *too long*, they are interrupted, and both teams are assigned to start working on task T5.

In our sample model we observe two possible on-the-fly decision scenarios, as described in the following:

a) If the duration of task T3/T4 performed by team B/A after the team has completed its preferably assigned task (the one with lower effort) is "too short" then interrupt it and start processing task T5 by both teams.

b) If the duration of the project is taking "too long" and it is currently processing non-vital tasks, then all of them are cancelled and both teams start working on T5.

Apparently, there is fuzziness in the project schedule description (i.e. "too short", "too long") that requires adequate modeling. For that we use the following fuzzy function (note that there is no limit as to what function can be used):

$$g(x) = \begin{cases} 0, x < a \\ \dfrac{x-a}{b-a}, a \le x \le b. \\ 1, otherwise \end{cases} \quad (2)$$

The exact distribution functions that we use in our sample model are shown in Table 3. For simplicity reasons we have limited them to normal and uniform distributions, which is not a limitation of the approach. In addition, to describe the fuzzy behavior we have used the function shown by Equation (2), with the following parameters:

1) "too short", $1 - g(x);\ a = 0.0, b = 5.0$
2) "too long", $g(x);\ a = 10.0, b = 15.0$

Both functions are illustrated in Table 4. In case (1) it means that the age intensity of the corresponding activity is less "too short" for 4.0 than e.g. for 2.0. Also, it means that it is extremely "too short" for the age intensity of 0.0, as well as it is not "too short" any more for the age intensity of 5.0. In case (2) the task has taken "too long" if its age is 15.0 at maximum, and the least "too long" if its age is 10.0.

Table 3. Input data for the sample project schedule (N-Normal, U-Uniform)

Features	Vital vs. NonVital	Task	Assigned Team and Duration Probability Distributions
Feature 1	Vital	T1	Team A: N(5.0, 1.0)
Feature 2	Vital		
Feature 3	Vital	T2	Team B: U(2.0, 10.0)
Feature 4	Vital		
Feature 5	Vital	T5	Team A and Team B: U(0.5, 2.0)
Feature 6	Vital		
Feature 7	Non vital	T3	Team A: U(2.0, 4.0)
			Team B: N(7.0, 1.2)
Feature 8	Non vital	T4	Team A: N(7.0, 1.0)
			Team B: U(2.0, 5.0)

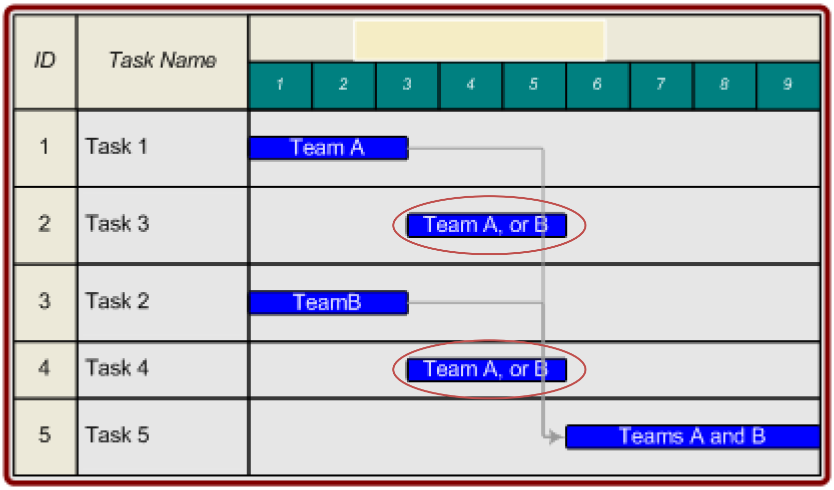
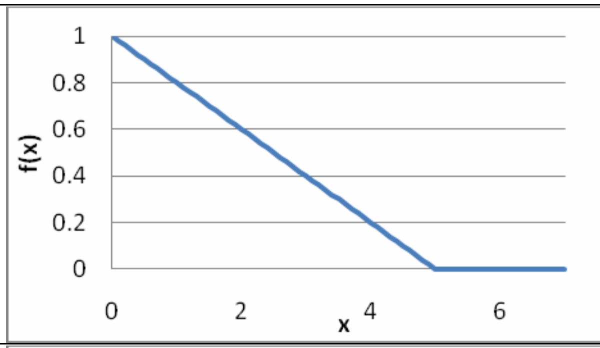
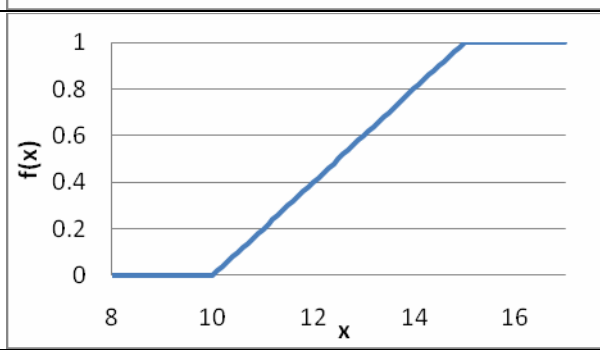


Fig. 3. Gantt chart of the example model, floating tasks are encircled in red color

Table 4. Fuzzy functions involved in the sample project schedule

"too short"	$a = 0.0,$ $b = 5.0$	
"too long"	$a = 10.0,$ $b = 15.0$	

5 Proxel-Based Simulation of Extended Project Schedules

In the following we provide the details of the proxel-based simulation of the sample project schedule that involves a floating task. This should serve as description of our approach through an example.

Each task in the model has a name, a priority level (vital or non-vital), duration probability distributions with respect to the possible team association, and a set of pre-requisite tasks. The proxel format of the state of the project schedule encompasses the following parameters:

- task vector $\{T_i\}$, where T_i is the task that team *i* is working on, or *I* for idle,
- age intensity vector $\{\tau_i\}$, for tracking the duration of tasks, and
- probability value.

Thus the format of the proxel is as follows:

$$Proxel = (Task\ Vector,\ Age\ Intensity\ Vector,\ Probability)$$

The initial proxel, i.e. the proxel that marks the initial state of the system would be ((T1, T2), (0, 0), 1.0). It describes the situation in which team A is working on task T1, and team B on task T2 with a probability of 1.0. In the next time step the model can do each of the following developments:

a) Task T1 completes,
b) Task T2 completes, or
c) None of the tasks completes.

Resulting into the following three proxels:

a) ((T3, T2), (0, Δt), p_1)
b) ((T1, T4), (Δt, 0), p_2)
c) ((T1, T2), (Δt, Δt), $1 - p_1 - p_2$)

In case (a), team A starts working on task T3, and also the corresponding age intensity is now reset to track the duration of T3. In case (b) team B takes over task T4. Case (c) shows the situation of both teams continuing what they have been doing before.

For demonstration, let us further develop the case (a). The next events that might happen are:

a1) Task T3 completes,
a2) Task T2 completes, or
a3) None of the tasks completes.

The interesting case is when T3 completes, which brings the model into state (T4, T2), where it subjected to various on-the-fly decision scenarios, as modeled by fuzzy functions. If team A has spent short time working on T4 (as it is considered non-vital), the task T4 is cancelled and both teams A and B proceed working on task T5. Else, team B waits for team A to complete task T4 before they proceed to task T5. For generating each new proxel, durations of tasks in progress need to be investigated for the decision modeling, as typically they are parameters of the fuzzy functions.

The state-transition diagram of the sample project schedule is shown in Fig. 4. As depicted with the extra-wide arrow , when team A is working on T4 and team B on T2 (state (T4, T2)), the transition associated with the completion of T2 depends on the time that team A has already spent on working on task T4. If it was “too long” then team B will stay idle and wait for its completion. On the other hand, if team A has just started working on task T4, then it is interrupted and both teams start working on task T5 which leads to completing the project. The same situation applies to the discrete state (T1, T3).

The red arrows in Fig. 4 shows the transitions due to Scenario (b), i.e. when the project currently processes non-vital tasks and its duration is too long. In that case there is a possibility that the tasks are interrupted and the project proceeds to task T5.

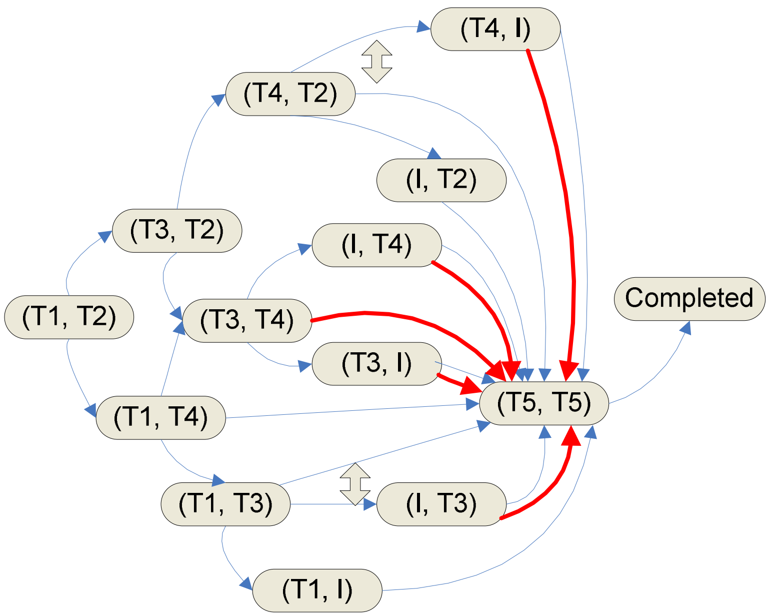


Fig. 4. State-transition diagram of the project schedule

The algorithm that we have developed represents an extension of the original proxel-based method [12, 13]. In particular, the differences can be summarized as:
Prior to processing each transition, check all possibilities for possible flow changes based on proxel parameters and corresponding to model description. Generate subsequent proxels correspondingly.

To illustrate the changes of the original algorithm, we illustrate the scenario of development from the proxel ((T4, T2), (t1, t2), p) in Fig. 5 using a proxel sub-tree to represent the specific elements. It is evident that in the probability calculation of the proxel, the fuzzy function $g(t)$ participates too. It increases the branching of possible paths that the model can take, as denoted by the green colored proxels.

Calculating probabilities of proxels need to include the fuzzy functions (here $g(\tau)$) associated with the possible model developments, as follows:

$$probability = \mu(\tau_1) * g(\tau_2) \tag{3}$$

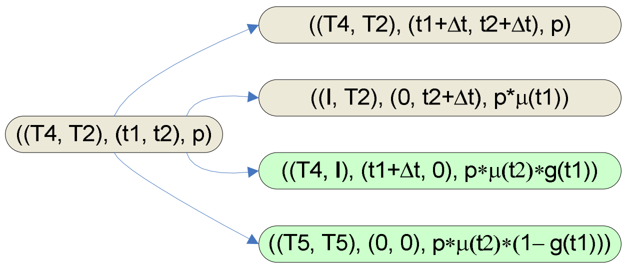


Fig. 5. Proxel sub-tree developed for the proxel ((T4, T2), (t1, t2), p)

```
INITIAL PROXEL
AddProxel  ((T1, T2, ( 0dt,  0dt)),  1.00000e+000)

STEP 1
AddProxel  ((T3, T2, ( 0dt,  1dt)),  1.48672e-007)
AddProxel  ((T1, T2, ( 1dt,  1dt)),  1.00000e+000)

STEP 2
AddProxel  ((T3, T2, ( 0dt,  2dt)),  2.43896e-007)
AddProxel  ((T1, T2, ( 2dt,  2dt)),  1.00000e+000)
AddProxel  ((T3, T2, ( 1dt,  2dt)),  1.48672e-007)

STEP 3
AddProxel  ((T3, T2, ( 2dt,  3dt)),  1.48672e-007)
AddProxel  ((T3, T2, ( 0dt,  3dt)),  3.96130e-007)
AddProxel  ((T1, T2, ( 3dt,  3dt)),  9.99999e-001)
AddProxel  ((T3, T2, ( 1dt,  3dt)),  2.43896e-007)

STEP 4
AddProxel  ((T3, T2, ( 0dt,  4dt)),  6.36983e-007)
AddProxel  ((T1, T2, ( 4dt,  4dt)),  9.99999e-001)
AddProxel  ((T3, T2, ( 2dt,  4dt)),  2.43896e-007)
AddProxel  ((T3, T2, ( 1dt,  4dt)),  3.96130e-007)
AddProxel  ((T3, T2, ( 3dt,  4dt)),  1.48672e-007)

STEP 5
AddProxel  ((T3, T2, ( 2dt,  5dt)),  3.96130e-007)
AddProxel  ((T3, T2, ( 4dt,  5dt)),  1.48672e-007)
AddProxel  ((T3, T2, ( 3dt,  5dt)),  2.43896e-007)
AddProxel  ((T3, T2, ( 0dt,  5dt)),  1.01409e-006)
AddProxel  ((T1, T2, ( 5dt,  5dt)),  9.99998e-001)
AddProxel  ((T3, T2, ( 1dt,  5dt)),  6.36983e-007)
```

Fig. 6. Output of the proxel-based simulation for the first 5 steps

and correspondingly for the alternative development:

$$probability = \mu(\tau_1) * (1 - g(\tau_2)). \quad (4)$$

This implies that if the fuzzy function $g(\tau)$ is defined on (a, b), then during the time that the age intensity τ_2 is within this interval, the number of possible developments of the model grows, and includes the ones modeled by the fuzzy function. This means that the state-space of the model is dynamically changing with respect to the state parameters.

The initial proxel development is shown in Fig. 6. This represents the verbose output of our program and it shows how the program works. As it displays only the first 5 steps and the time step $\Delta t = 0.1$ there are only two discrete states that the model can be in: (T1, T2) and (T3, T2). This is so because the completion of task T2 is uniformly distributed from 2.0 to 10.0, thus it cannot transit to (T1, T4) before the 20th time step.

6 Experiments and Results

6.1 Experimental Environment

The experiments were run on a standard workstation with an Intel Core2Duo Processor at 2.0 GHz and 1 GB RAM. The choice for Δt was 0.1 and the simulation was run up to time $t = 25$. This implies that the number of simulation steps was 200.

The computation time for this experiment was ca. 5 seconds. In the following we present the results, i.e. the statistics that were calculated during this simulation experiment. The input data is provided in Table 3.

6.2. Experiments

The goal of the experiments is to show the importance of modeling the effects of on-the-fly human decision behaviors on project schedules. For that purpose we first simulated the project schedule in an ideal scenario, i.e. excluding any intrusions during project running. Next, we simulated the project duration exposed to the hypothetical scenarios (both (a) and (b)) for on-the-fly project flow decisions. To study the effect of neglecting them, we compare both solutions and present the results with a chart.

In Fig. 7, we observe the probability distribution of project duration for all combination of on-the-fly decision scenarios. It is evident that in the ideal case, i.e. the case that is not a subject to various on-the-fly decisions about the workflow, the project duration is the longest, as expected. However it is very interesting that the difference in duration is more than 5 time units (in this case more than 25% of the duration of the project). In the ideal case the project completes with a probability of 1.0 in ca. 20 time units. In the case with both scenarios this duration is ca. 13 time units, and in the case with only scenario (a) it is ca 18. Therefore, the difference is significant.

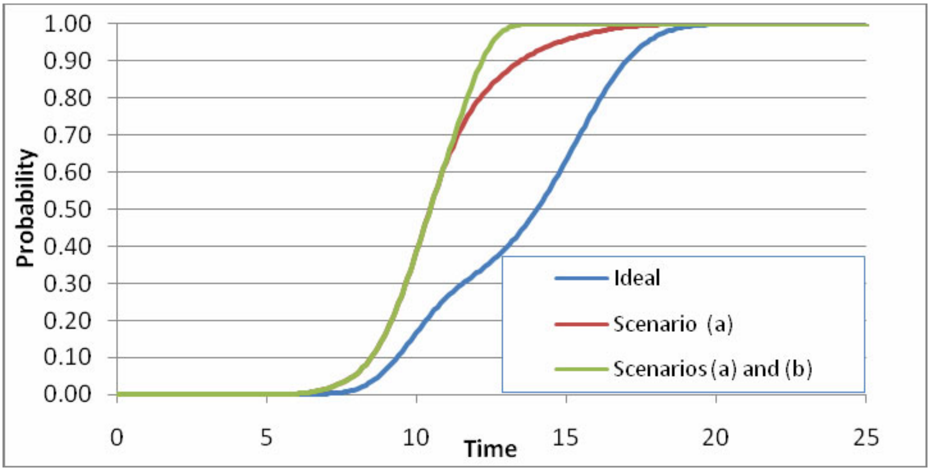


Fig. 7. Probability distribution of the duration of the project schedule with the 3 possible combinations of scenarios

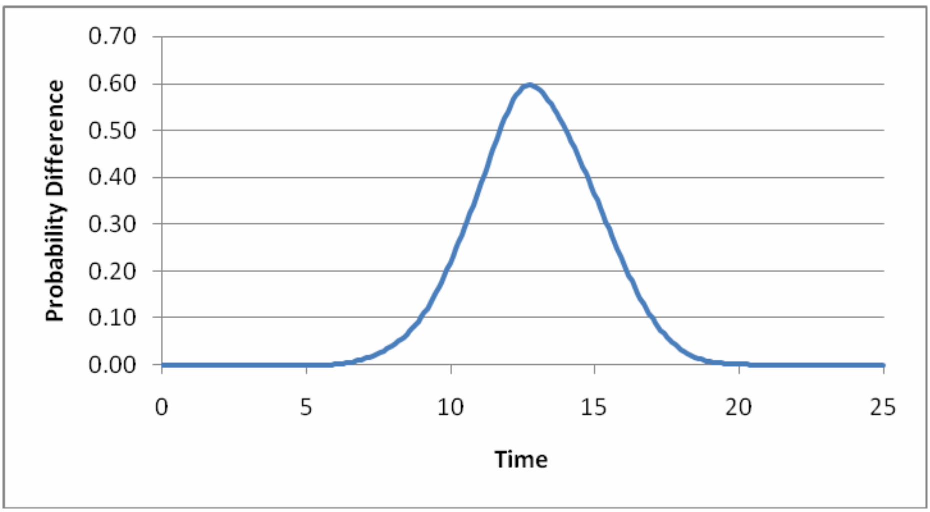


Fig. 8. Project schedule duration probability distributions difference for the ideal and project schedule with both scenarios

In order to represent closely the effect of the modeling and simulation of the on-the-fly workflow decisions, Fig. 8 shows the difference of the two probability distributions, with and without on-the-fly decision scenarios. The results show that in our sample model, it goes up to ca. 0.6, which is a very significant probability difference.

7 Discussion

The approach that we presented allows a higher degree of uncertainty in project schedules to be modeled and simulated. The uncertainty that we observe is in terms of duration of tasks, task allocation, as well as arbitrary on-the-fly decisions that influence the workflow. We all witness that these things happen almost every time and in every project. Thus, simulation models need to consider them in order to obtain accurate measures for the duration of project schedules. Very often, these factors are neglected, and by our example we showed what difference they can make.

In our example model, the probability difference for the completion of the project reached ca. 0.6, and this is still just a toy model. In real project schedules it can be more extreme and thus it has to be taken into account. The question that arises is how to obtain the numbers that represent and model these behaviors. We believe that they can be modeled by historical data and tracking from previous projects of similar types. In addition, expert knowledge and common sense can help to a great extent.

8 Summary and Outlook

This paper presents a more realistic project schedule simulation and modeling approach that allows for a high level of uncertainty. The purpose of this simulation model is: (a) to model the uncertainty of human resources allocation to the different project tasks and (b) to take advantage of this uncertainty to simulate various on-the-fly human decisions and study their impact on the project duration.

To extend our work we plan to address the effect of these uncertainty factors on the productivity and budget, by adding value, effort and cost parameters. In addition, we intend to extend our simulation model to handle the effects of requirements volatility in software engineering.

References

1. Joslin, D., Poole, W.: Agent-based simulation for software project planning. In: Proceedings of the 37th Conference on Winter Simulationl 2005, Orlando, Florida (2005)
2. Jing-wen, Z., Hui-fang, S.: Multi-Mode Double Resource-Constrained Time/Cost Trade-Offs Project Scheduling Problems. In: International Conference on Management and Service Science, MASS 2009 (2009)
3. Huang, W., et al.: Project Scheduling Problem for Software Development with Random Fuzzy Activity Duration Times. Springer, Heidelberg (2009)
4. Herroelen, W., Leus, R.: Project scheduling under uncertainty: Survey and research potentials. European Journal of Operational Research 165(2), 289–306 (2005)
5. Arauzo, J.A., et al.: Multi-agent technology for scheduling and control projects in multi-project environments. An Auction based approach. Inteligencia Artificial 42, 12–20 (2009)
6. Sobel, M.J., Szmerekovsky, J.G., Tilson, V.: Scheduling projects with stochastic activity duration to maximize expected net present value. European Journal of Operational Research 198(3), 697–705 (2009)
7. AbouRizk, S.M., Wales, R.J.: Combined discrete-event/continuous simulation for project planning. Journal of Construction Engineering and Management 123(1), 11–20 (1997)

8. Perminova, O., Gustafsson, M., Wikström, K.: Defining uncertainty in projects–a new perspective. International Journal of Project Management 26(1), 73–79 (2008)
9. Rauch-Gebbensleben, B., Dammasch, K., Horton, G.: Generierung und Visualisierung des Zielkorridors in Projektplänen mit stochastischen Einflussgrößen. In: Simulation und Visualisierung 2008, Magdeburg (2008)
10. Isensee, C., Horton, G.: Proxel-Based Simulation of Project Schedules. In: European Simulation Multiconference (2004)
11. Pfahl, D., Lebsanft, K.: Using simulation to analyse the impact of software requirement volatility on project performance. Information and Software Technology 42(14), 1001–1008 (2000)
12. Horton, G.: A new paradigm for the numerical simulation of stochastic Petri nets with general firing times. In: Proceedings of the European Simulation Symposium (2002)
13. Lazarova-Molnar, S.: The Proxel-Based Method: Formalisation, Analysis and Applications. In: Faculty of Informatics. University of Magdeburg, Magdeburg (2005)
14. Stewart, W.J.: Introduction to the Numerical Solution of Markov Chains. Princeton University Press, Princeton (1994)
15. Cox, D.R.: The analysis of non-Markovian stochastic processes by the inclusion of supplementary variables. Proceedings of the Cambridge Philosophical Society 51(3), 433–441 (1995)
16. Lin, F.J., Chu, P.M., Liu, M.T.: Protocol verification using reachability analysis: the state space explosion problem and relief strategies. ACM SIGCOMM Computer Communication Review 17(5), 126–135 (1987)
17. Isensee, C., Horton, G.: Proxel-Based Simulation of Project Schedules. Citeseer (2004)
18. Leffingwell, D., Widrig, D.: Managing Software Requirements: a use case approach. Pearson Education, London (2003)

Author Index