FSM-Based Object-Oriented Organization Modeling and Simulation

Vojtěch Merunka[1,2]

[1] Czech University of Life Sciences Prague, Faculty of Economics and Management, Department of Information Engineering

[2] Czech Technical University in Prague, Faculty of Nuclear Sciences and Physical Engineering, Department of Software Engineering in Economy

vmerunka@gmail.com

Abstract. This paper presents the idea of the convergent approach to modeling and simulation of business requirements and software development based on the combination of the FSM and the Object-Oriented Approach. The first part of this paper discusses the motivation for the need to connect two areas of modeling and simulation: business requirements and software engineering. The second part of the paper presents the idea of modeling of processes and business situations as more mutually associated FSM representing particular objects and shows examples of this approach. The third and the last part of the paper presents the mapping of the proposed approach to BPMN-based and UML-based models and provides interesting new findings resulting from the proposed approach. This approach is based on our experience with our recent practical projects concerning business modeling and simulation in various application areas (e.g. health care, gas supply industry, regional management, administration process design of a new faculty of a university) and subsequent software development in these application areas.

Keywords: business process modeling, conceptual modeling, organizational modeling and simulation, object-oriented approach, FSM, BPMN, UML, BORM.

1 Introduction

Software application development for business and similar domain-specific areas shifts the attention at the requirement analysis and design activities, e. g. from the programming level to the modeling level. MDA (Model-Driven Architecture) [11] is the recent approach based on strategy of the application development based on requirements, conceptual and design modeling. The typical tool used in this area is the UML - Unified Modeling Language [19].

Our idea of continuous model-driven engineering aims to fill in the gap between of two worlds of "Business", which is process-based and requires deep management and economical knowledge, and "IT", which uses its own modern software development tools and techniques. This is to minimize the failure rate of information systems through the application of proper simulation and modeling techniques before the system is built. We want to advance the discipline of conceptual modeling in the area between the use of business domain knowledge and the use of modern advanced programming techniques and tools such as object-oriented programming environments (.NET, XCode, VisualWorks, ...), prototyping environments (Self, Squeak), non-traditional programming languages (Smalltalk, Objective-C) etc.

The goal of our paper is to converge the BPMN and UML modeling using approach, which will enable to use only one modeling and simulation paradigm trough the entire software system development life-cycle. Our paper contributes to the area of system modeling methodologies, tools and techniques to enable simulation, verification and validation activities.

2 Motivation

2.1 Our Experience

In our experience, any modeling and simulation tool and diagramming technique used at this kind of business projects should be comprehensible to the stakeholders, many of whom are not software engineering literate. Moreover, these diagrams must not deform or inadequately simplify requirement information. It is our experience that the correct mapping of the problem into the model and subsequent visualization and possible simulation is very hard task with standard diagramming techniques. 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 (as well as BPMN) lost this power of simplicity and clarity.

That is reason why we developed and successfully used our own BORM process diagraming technique [10] and our own way to start business system analysis. The initial work on BORM (Business-Object Relation Modeling) was carried out in 1993 under the support of the Czech Academic Link Programme (CZALP) of the British Council, as part of the VAPPIENS[1] research project; further development and recent practical projects in the last decade has been carried out with the support of Craft.CASE Ltd. - the British software consulting company supporting innovative technologies. (*VAPPIENS was funded by the British Governments CZALP, 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 in last 15 years for a number of business consulting and software engineering projects including

- the identification of business processes in metropolitan hospital,
- the modeling of properties necessary for the general agricultural commodities wholesale sector requested by the Agrarian Chamber,

[1] Visual Application Programming Paradigms for Integrated ENvironmentS.

– as a tool for business process reengineering in the electricity supply and gas supply industry (see example in figure 1),
– as a tool for business process reengineering for telecommunication network management,
– in organizational modeling and simulation of regional management project concerning the analysis of the legislation and local officials' knowledge such as living situations, law, urban planning etc. (see simulation example in figure 2).
– several business process simulation projects in area of simulation of marketing chains for Makro,
– visualization of safety and fire regulations in the electric power engineering sector and
– administration process design of a new Faculty of Information Technology of the Czech Technical University in Prague.

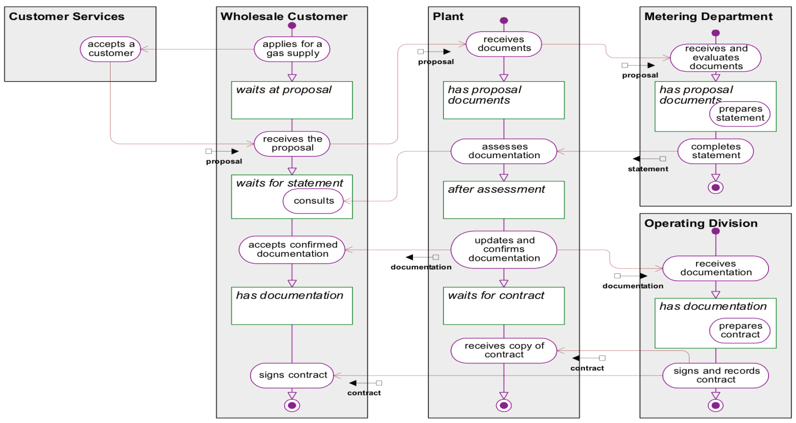

Fig. 1. Workflow process of contract on gas supply

However during the last decade there was an significant upgrade of UML, and also a new standard for business process modeling BPMN has been developed and we recognized that our approach is close to both of them. We think that based on our past experience, we can propose a new approach. This new approach is based on the object-oriented concepts, is using the well proven technology of FSM for modeling and simulation. It has almost the same expressive power as the UML and the BPMN together, but it is expressed in a uniform and simpler manner.

2.2 Motivation - The Gap between the World of Business Modeling and Software Modeling

Nowadays, there is a great variety of tools and techniques for business modeling. Unfortunately, there is no standard for business modeling like UML for software

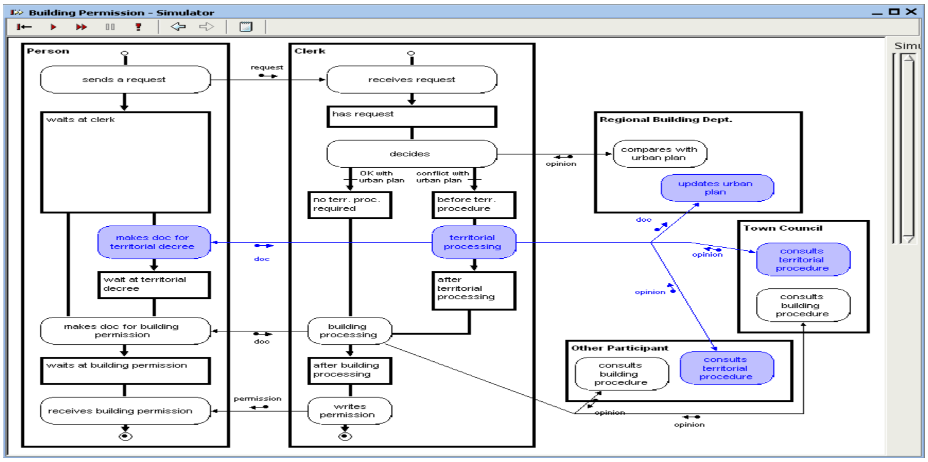

Fig. 2. Simulation of the regional management process on building permission

engineering yet. Nevertheless, we can presume that the approach to become a standard will be the BPMN. [2,17,7] In Europe, Aris and its EPC diagram is still very popular; however BPMN authors say this technique becomes obsolete and is almost unknown outside the Europe.

When OOP started to be used in practice, it was assumed that object-oriented technology will become mainstream in software development (which was correct) but also that object-oriented approach will affect the approach to business process modeling, organizational engineering and the whole area of activities preceding the formulation of information system requirement. That, unfortunately, did not happen. These pre-implementation activities are still carried out the old fashioned way, which results in a semantic gap between the world of business modeling and the world of software modeling as shown in figure 3. Latest publications even express opinions such as "OOP has failed, OOP is a dead end, etc.". These are written by people who have no experience with pure object-oriented languages and environments, but only with hybrid ones (e.g. Delphi, Java, etc.) and they generalize their negative practical experience with failed projects to the whole paradigm.

Expected output of the business engineering activities is information or data in a form that can be directly used as an input for implementation of the system in the spirit of software engineering. However, this is not the easy case; there are following issues described by Illgen and Hulin in [9] and Van der Aalst in [1]:

1. *oversimplification* - while trying to at least finish business and organizational model we are forced simplify the problem being modeled and
2. *inability* - some important details cannot be recorded because of the method being used.

We believe together with Schach [16] that this is the reason why software system modeling and subsequent design of business applications is a pretty hard deal: Today's world of software development still works with algorithmic and imperative style of thinking, and therefore the produced models concentrates on functions, function separation, continuity etc. But this "behavioral" and "straight timeline" approach goes against the business modeling paradigm which mainly focuses on states, situations, rules - often even in a form of statutory regulations and directives and concurrency. Behaviors are of less importance and make sense only if we know what they are good for. That is why it is much more natural to start analyzing a business process from the description of initial situation (e.g. we have goods, we want to sell it, the customer wants our goods and has money, ...) and the description of the desired final situation (e.g., the paying customer makes us a good profit). In this perspective, the sequence of behaviors is only a result of an effort to get from the initial to the final situation, thus, being much more declarative then imperative style of modeling. The significance of the declarative approach is expected to grow even in the software programming itself, where the imperative (e.g. behavior-oriented) modeling style is still prevalent, although unnatural for business and organizational modeling.

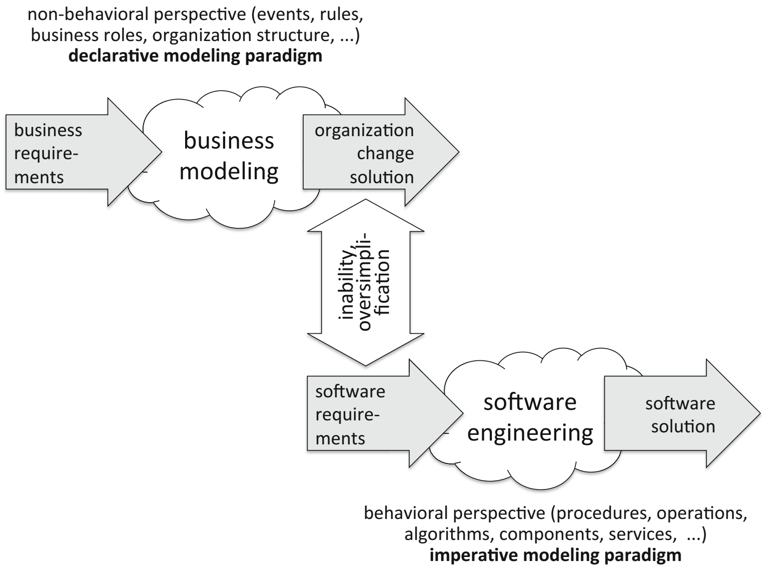
non-behavioral perspective (events, rules, business roles, organization structure, ...)
declarative modeling paradigm
business require-ments
business modeling
organization change solution
inability, oversimpli-fication
software require-ments
software engineering
software solution
behavioral perspective (procedures, operations, algorithms, components, services, ...)
imperative modeling paradigm

Fig. 3. Semantic gap

2.3 Gap between Software Modeler and Software Programmer

This second gap persisting from 1970's is known as the impedance problem. It concerns the different interpretation of conceptual relations by the analysts and by the programmers. This gap is caused by different interpretations of ontological categories of the supertype-subtype relationship (inheritance), whole-part relationship (composition) and association relationship by various kinds of programmers and developers. This problem has significant influence to the modern application development in case of so called object-relational impedance mismatch (mentioned by Hohenstein in [8], for example), where relational databases on the server side stress association relationship (in fact the only native relationship supported by the relational data model), but client-side applications are based on the object-oriented programming languages, in which we can work with the inheritance and compositions, but we don't have any direct language support for associations (see table 1). This leads to the need to permanently transform data from their server data model to their client model and vice versa. In this situation, it is pretty hard for analysts to maintain the correct usage of associations and compositions in models they need to deal with, as the essential difference between these two relationship concepts is usually eclipsed by their implementation in a programming environment.

Table 1. Impedance problem

	object-oriented programming languages (client side)	relational databases (server side)
inheritance (is-a relationship)	YES	NO
compositions (whole-part relationship)	YES	NO
associations	NO, must be implemented via compositions	YES

3 Our Solution - FSM and OOP Combination

Our solution is based on the reuse of old thoughts from the beginning of 1990s regarding the description of object properties and behavior using finite state machines (FSM). The first work expressing the possible merge of OOP (Object-Oriented Paradigm) and FSM was the Shaler's and Mellor's book [14]. One of the best books speaking about the applicability of OOP to the business modeling was written by Taylor [18]. These works together with our practical experience is why we believe that the business requirement modeling and simulation and software modeling could be unified on the platform of OOP and FSM.

Figure 4 shows and example of a model of a book in a library represented in a form of a finite state machine with three states: a book on a shelf, a book on loan and a returned book to be put back on a shelf. These states are easily recognizable through an interview with domain experts. When these states are recognized, it is possible to identify behaviors required for transferring books between the states. Of course, other objects in the system can be modeled this way too. Let us take a borrower as an example. Even him has his own states which can be related to the states of the book; and analogically, the borrower's behaviors can be related to the book's behaviors, see figure 5. This figure 5 shows that a satisfied borrower is associated with an object of a book on loan in

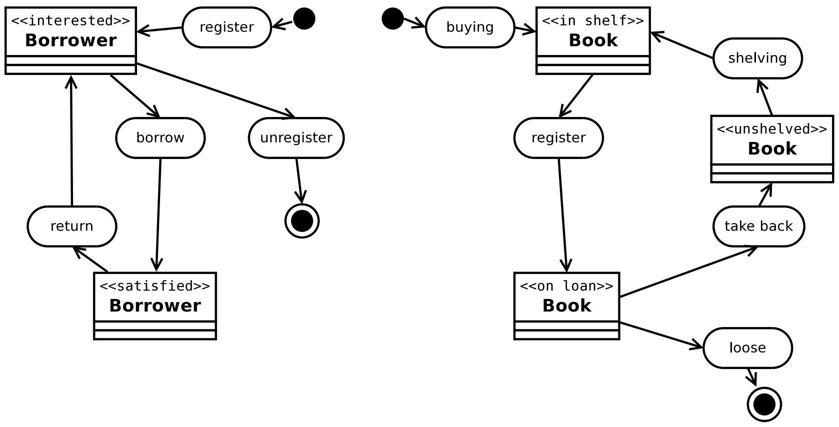

Fig. 4. A book and a borrower as FSM

a relationship with cardinality of a 1 to many. That means, that every book on loan has to have its borrower and every satisfied borrower is satisfied if and only if he has at least one book borrowed. This is the possible business situation of our model. Similarly, we can identify the associations between borrower interested in borrowing a book and available books in the shelves. This association can be another possible business situation of our model.

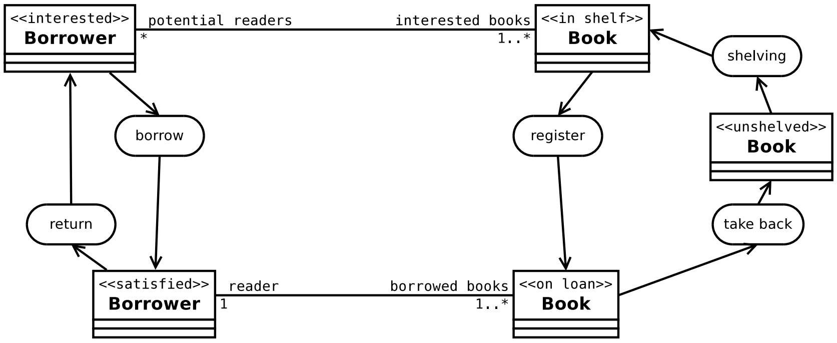

Fig. 5. Relationships (associations) between a book and a borrower

Moreover, the notion of association between states of objects can interestingly be generalized for the links between behaviors. It is the link between the activity borrowing and activity registration in the figure 6. This new kind of relationship is analogous with data association and can be regarded as the generalized message passing already existing in dynamic UML models.

This model can be further processed. One option is to further refine or conversely simplify (aggregate) the detail of the model. Figure 7 gives an example of

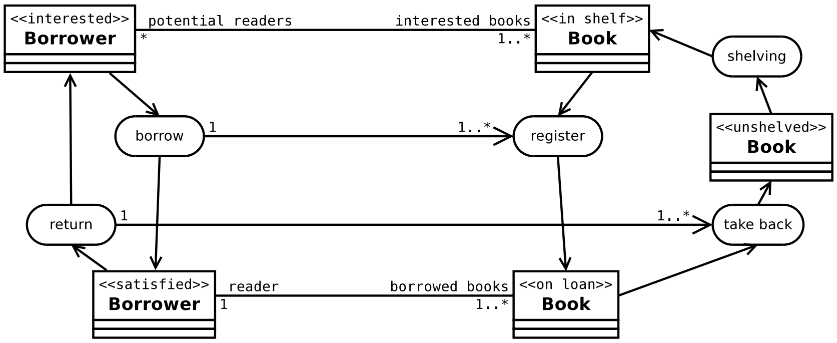

Fig. 6. Communication links between activities of a borrower and activities of a book

two levels of detail of the same model. The lower model is the same model as in figure 5, but the one at the top is its simplified - aggregated - and very familiar form. It is a common model of a data structure, part of the UML class diagram, which shows a borrower and a book connected with a relationship-association (* to 1..*). This figure also illustrates that this relationship can be built as a union of two more detailed situations from the model at the bottom. For this act of refinement or aggregation we can conveniently use the well known conceptual relationship of supertype-subtype or whole-part (composition).

Besides changing the level of detail of models, we can also filter these object-oriented FSM models. Filtering is an operation of simply removing an unnecessary detail. In the figure 8 there is a model with removed state returned book and related behaviors, because the borrower has not any direct connection to these parts of the model and therefore they are unnecessary from his point of view.

It is obvious that the operations of filtration and simplification-specification can be combined. In the figure 9, the borrower is simplified into a single object class with no states and a book has only two states being important for borrowers.

4 Discussion

The presented modeling approach unifies UML-style object modeling and business process modeling in the business and organizational engineering techniques and tools style. Models like UML, BPMN and other can be easily derived from this model. Overall schema of this unified modeling approach is laid out in the figure 10.

From the viewpoint of this schema, we can consolidate the terms used in modeling with BPMN and UML into a table 2.

The proposed unified approach generalizes object modeling. In this perspective, modeling in UML and BPMN could be perceived as mutually following stages of the new more universal approach according to the MDA principles

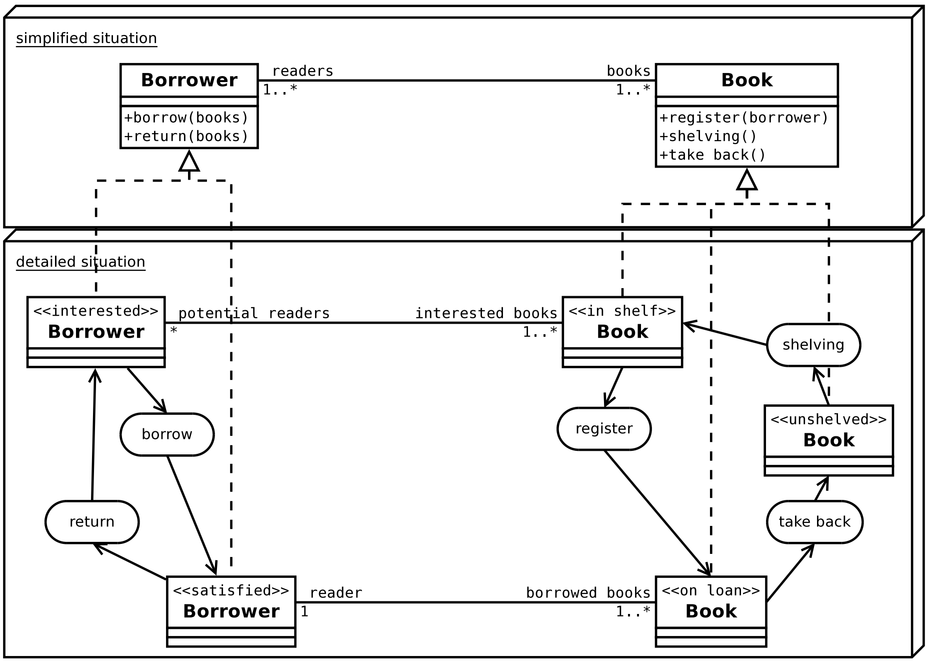

Fig. 7. Simplified and detailed view of the same model

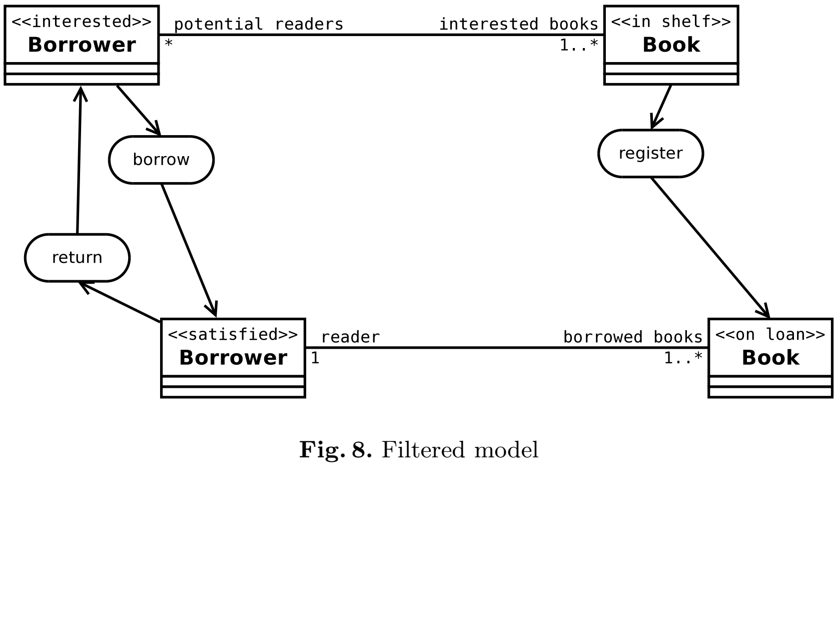

Fig. 8. Filtered model

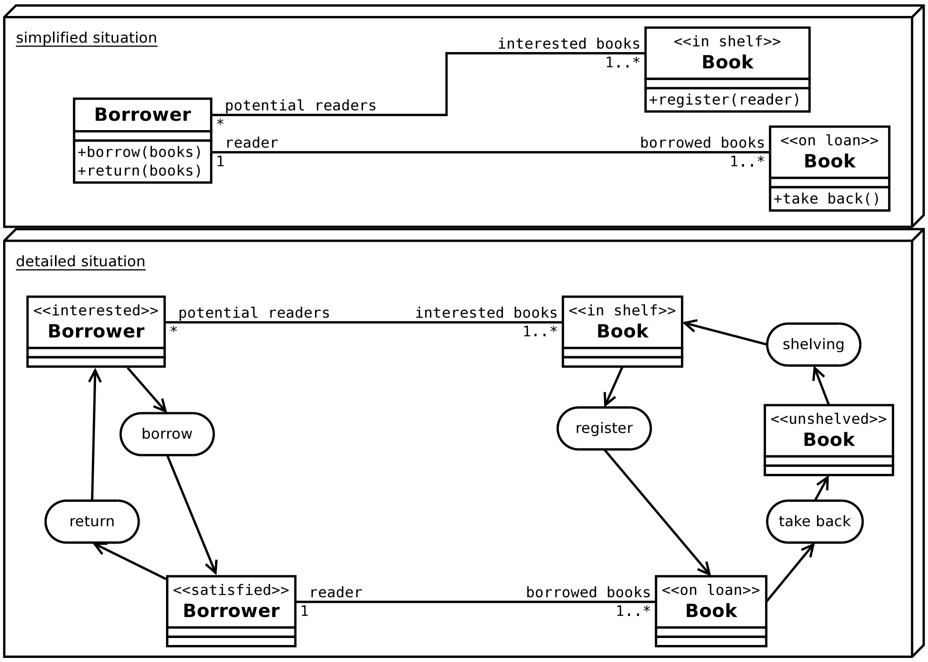

Fig. 9. Simplified model using both filtration and generalization

[11]. Figure 11 shows mapping or our model to the standard BPMN and figure 12 shows the same model expressed in the state-chart of the standard UML. Possible advantages of our convergency approach follows:

1. BPMN and UML both cover only a *subset* of the entire exploitable space of modeling concepts (see table 2).
2. The most important concepts are *states* of objects. Behaviors represent only the necessary glue between them. Both business processes and software components should be therefore modeled by starting with their states - *situations* of participating objects in the requested structure in some time. Modeling can be easier, more precise and less behavioral-imperative then it is today.
3. Our approach also denotes interesting *conceptual difference between compositions and associations*. The composition represents a greater detail of the same object, but the association is an information relationship between two different objects. This differentiation is very valuable especially because programmers often mix up these two terms. From a certain point of view, the composition can be seen similar to IS-A relation (in order to express detail of the concrete object) rather than to association which is really about information relationship between different objects. Mode detail about these conceptual differences is described in [13].
4. A new modeling term was defined - *communication* - which de facto generalizes the OOP term message. Communication is a link between behaviors

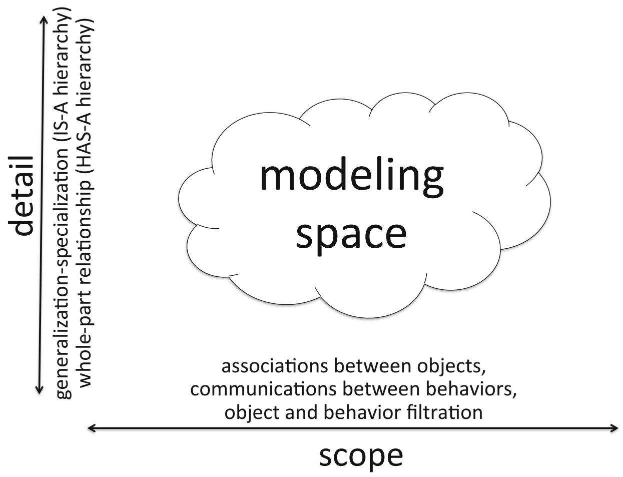
modeling
space
detail
generalization-specialization (IS-A hierarchy)
whole-part relationship (HAS-A hierarchy)
associations between objects,
communications between behaviors,
object and behavior filtration
scope

Fig. 10. Modeling space

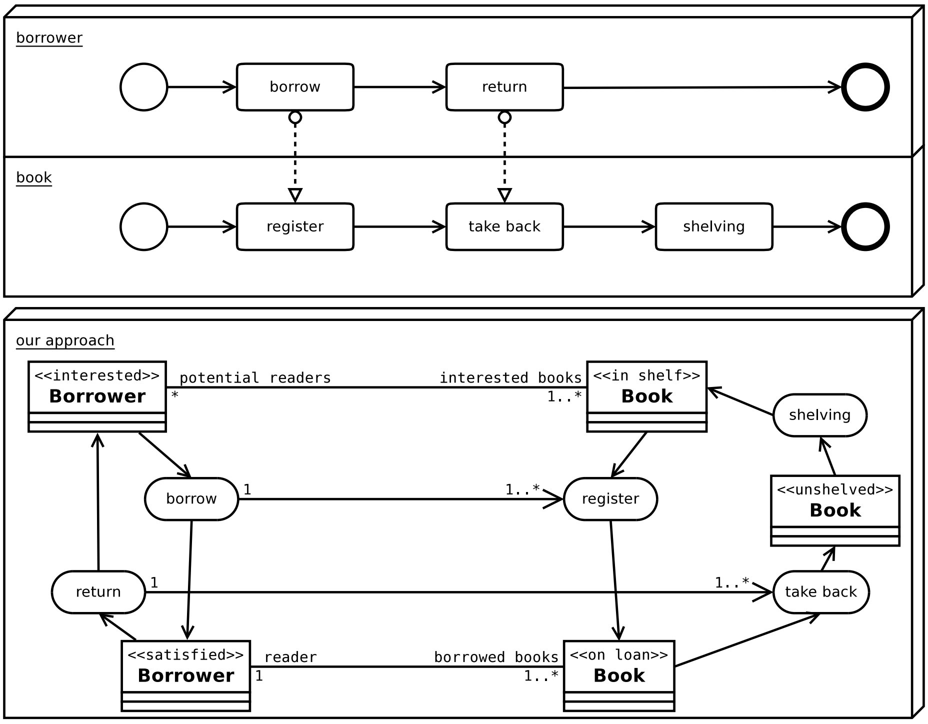

Fig. 11. Mapping to BPMN

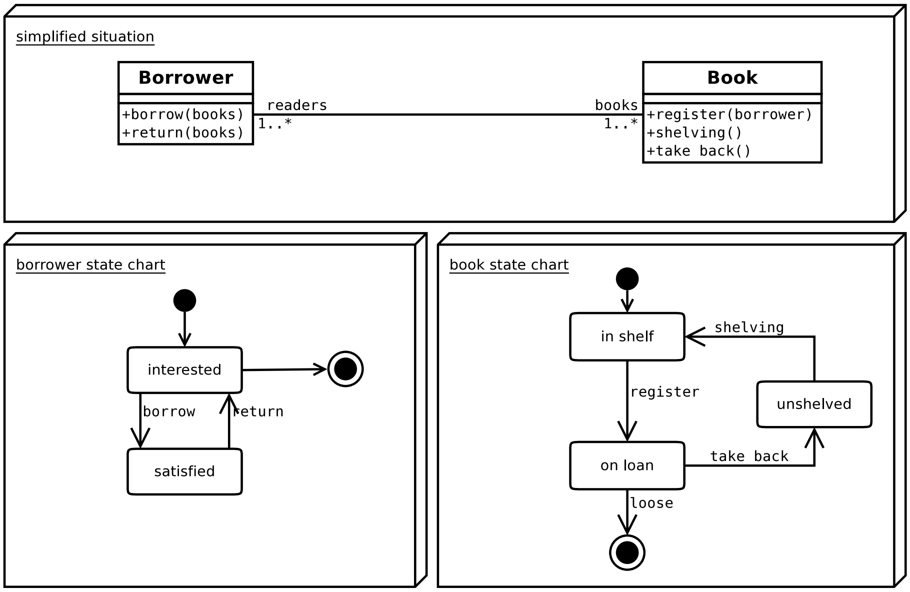

Fig. 12. Mapping to UML

Table 2. Modeling Concepts Coverage

our approach	BPMN-based approach	UML-based approach
objects	swimlines only (process participants)	YES
object states	can be expressed by various events	YES
associations between objects (data links)	NO	YES
associations between object states (data links)	YES	NO
object behaviors (activities)	YES	YES
communications between behaviors (messages)	YES	YES
generalization-specialization relationship (inheritance)	NO	objects only
whole-part relationship (composition)	NO	objects only

of different objects whose states could be also associated. This term communication is symmetric to the term association and can have own cardinality as well (although we would recommend using the term "frequency"). Association is an information relationship between objects or their states, whereas communication is an analogous relationship between the behaviors of these objects.

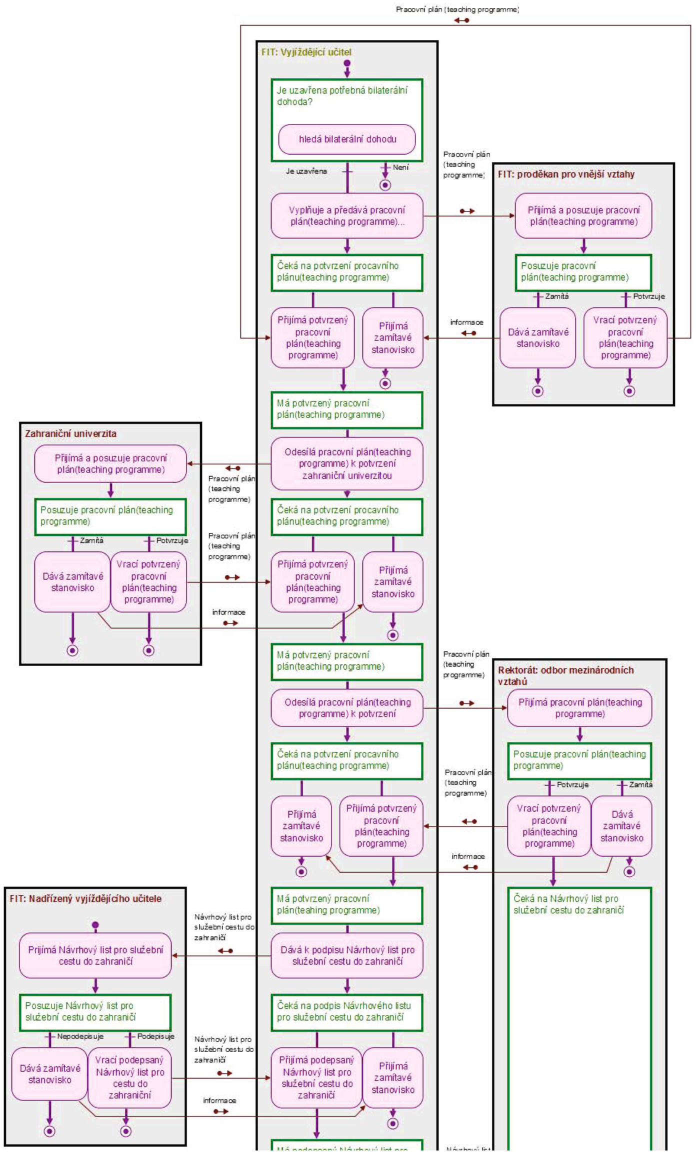
Pracovní plán (teaching programme)
FIT: Vyjíždějící učitel
Je uzavřena potřebná bilaterální dohoda?
hledá bilaterální dohodu
Je uzavřena
Není
Vyplňuje a předává pracovní plán(teaching programme)...
Čeká na potvrzení procavního plánu(teaching programme)
Přijímá potvrzený pracovní plán(teaching programme)
Přijímá zamítavé stanovisko
FIT: proděkan pro vnější vztahy
Přijímá a posuzuje pracovní plán(teaching programme)
Posuzuje pracovní plán(teaching programme)
Zamítá
Potvrzuje
Dává zamítavé stanovisko
Vrací potvrzený pracovní plán(teaching programme)
informace
Má potvrzený pracovní plán(teaching programme)
Odesílá pracovní plán(teaching programme) k potvrzení zahraniční univerzitou
Zahraniční univerzita
Rektorát: odbor mezinárodních vztahů
Odesílá pracovní plán(teaching programme) k potvrzení
Přijímá pracovní plán(teaching programme)
FIT: Nadřízený vyjíždějícího učitele
Prijímá Návrhový list pro služební cestu do zahraničí
Posuzuje Návrhový list pro služební cestu do zahraničí
Nepodepisuje
Podepisuje
Vrací podepsaný Návrhový list pro cestu do zahraniční
Návrhový list pro služební cestu do zahraničí
Dává k podpisu Návrhový list pro služební cestu do zahraničí
Čeká na podpis Návrhového listu pro služební cestu do zahraničí
Přijímá podepsaný Návrhový list pro služební cestu do zahraničí
Čeká na Návrhový list pro služební cestu do zahraničí
Má potvrzený pracovní plán(teahing programme)

Fig. 13. ERASMUS Student Mobility for Studies - a fragment of a real simulation model

This strong difference between the concept of association (e.g. the relationship between different objects on the same level) and composition (e.g. the tool for making more or less detail of some particular object) can help the software developers to solve the impedance problems.

Figure 13 brings an example of a real process. This is the fragment of the process ERASMUS Student Mobility for Studies made for the new Faculty of Information Techology of the CTU Prague. There was total number of 63 similar business processes and 55 participants (objects), where each of them has been modelled as a FSM and connected in various ways with other FSM. Average number of interconnected FSM in one process was 7 and the total number of micellaneous data flows (e.g. documents, forms, e-data, ...) running between particular FSMs was 72.

5 Conclusion

In this paper we presented the idea of the convergent approach to modeling and simulation of business requirements and software development. Our approach called BORM is combining the object-oriented approach and finite-state machines and is based on our practical experience with recent project, which were aimed to help the teams made by business consultants and software developers from various areas (e.g. health care, gas supply industry, regional management, university management). We feel that the highest value of our approach is generated by the way of modeling, which smoothly connects two different worlds: business engineering and software engineering. We believe that this approach can help in future possible integration of BPMN and UML models for complex projects requiring the strong collaboration between software system architects and problem domain experts in area of organization structures modeling and subsequent simulation as it is predicted by Scheldbauer in [15].

Our future work will be focused on implementation of the proposed concepts in selected CASE tools and on incorporation of this approach into the BORM methodology [10]. The hot candidate for this project is the MetaEdit CASE tool by the Finnish company Metacase Ltd. [12] The autor would like to acknowledge the support of the research grant SGS11/166/OHK4/3T/14 and NAKI MK-S-3421/2011 OVV.

References

1. van der Aalst, W.: Business Process Simulation Revisited. In: Keynote Speech at the EOMAS Workshop (2010), `http://www.eomas.org` (cit: April 10, 2011)
2. Allweyer, T.: BPMN 2.0. Books on Demand GmbH, Norderstedt (2010) ISBN 978-3-8391-4985-0
3. 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)
4. Barjis, J., Reichgelt, H.: A Petri Net Based Methodology for Business Process Modeling and Simulation. In: The 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
5. Degen, W., Heller, B., Herre, H., Smith, B.: GOL - Towards an axiomatized upper level ontology. In: Proceedings of FOIS 2001, Ogunquit, Maine, USA. ACM Press (2001)

6. Eriksson, H., Penker, M.: Business Modeling with UML. John Wiley and Sons (2000) ISBN 0-471-29551-5
7. Grosskopf, A., Decker, G., Weske, M.: Business Process Modeling Using BPMN. Meghan Kiffer Press (2006) ISBN 978-0-929652-26-9
8. Hohenstein, U.: Bridging the gap between C++ and relational databases. In: Cointe, P. (ed.) ECOOP 1996. LNCS, vol. 1098, pp. 398–420. Springer, Heidelberg (1996)
9. Ilgen, D., Hulin, C.L.: Computational Modeling of Behavior in Organizations - The Third Scientific Discipline. American Psychological Association, Washington, DC (2000) ISBN 1-55798-639-8
10. 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
11. MDA – The Model Driven Architecture, OMG – The Object Management Group, `http://www.omg.org`
12. MetaCase - Domain-Specific Modeling with MetaEdit+, `http://www.metacase.com`
13. Molhanec, M.: Conceptual Normalisation Formalised. In: Barjis, J., Eldabi, T., Gupta, A. (eds.) EOMAS 2011. LNBIP, vol. 88, pp. 159–172. Springer, Heidelberg (2011)
14. Shlaer, S., Mellor, S.: Object Lifecycles: Modeling the World in States. Yourdon Press (1992) ISBN 0136299407
15. Scheldbauer, M.: The Art of Business Process Modeling–The business Analyst Guide to Process Modeling with UML and BPMN. Cartris Group, Sudbury MA (2010) ISBN 1-450-54166-6
16. Schach, S.: Object-Oriented Software Engineering. McGraw Hill, Singapore (2008) ISBN 978-007-125941-5
17. Silver, B.: BPMN Method & Style. Cody-Cassidy Press, Aptos (2009) ISBN 978-0-9823681-0-7
18. Taylor, D.A.: Business Engineering with Object Technology. John Wiley (1995) ISBN 0-471-04521-7
19. The UML standard, OMG – The Object Management Group, ISO/IEC 19501, `http://www.omg.org`