# Bridging the gap between service-oriented and object-oriented approach in information systems development

**Viera Rozinajová, Marek Braun, Pavol Návrat and Mária Bieliková**

**Abstract** Service oriented architecture is nowadays one of the most promising concepts in the area of information systems development. The advantages of this approach are obvious. Considering software development methodologies, the most popular ones in the last decade were based on object-oriented techniques. The goal of this paper is to investigate the possibilities of extending object-oriented methodology for information systems development by service oriented approach as well as to examine benefits, which it brings. We propose an augmentation of the object-oriented methodology BORM in order to incorporate service-oriented techniques into information systems development. To demonstrate advantages of our approach, an experimental life cycle of information system development was performed and its features evaluated.

## 1 Introduction

Service oriented architecture (SOA) is today the most challenging concept in information systems development. Indeed, there are reasons to believe that it brings flexibility, scalability and faster system development (Bell 2008). At the time being, information systems are not prepared for quick and dynamic modifications. The changes in processes having impact on entire organization are reflected in the systems with rather long time delay – if ever. If the organizations want to be successful, they have to face up this problem.

In the SOA the system consists of modular components with standardized interfaces through which components (services) provide their functionality. The ap-

Institute of Informatics and Software Engineering
Faculty of Informatics and Information Technologies
Slovak University of Technology, Ilkovicova 3, 842 16 Bratislava, Slovakia
{rozinajova, navrat, bielik}@fiit.stuba.sk

proach is rooted in traditional software development best practices and standards. However, the idea of properly bridging new and old software technologies is a novel one. To set up SOA, it is often necessary to concentrate on business processes, their modeling and management. According to (Erl 2008) many of the services that will eventually be modeled and designed will be business services, responsible for accurately encapsulating and expressing business logic. Therefore, a comprehensive set of business models and specifications is needed. In this paper we focus our attention to the systems that are based on the business process-oriented approach to SOA, i.e. the systems, consisting of the services that support business process realization.

In case the services express business logic they are abstract representations of business capabilities that the business can compose into processes. Therefore a well defined business process is a critical success factor for deploying SOA and using a methodology based on business process modeling would be beneficial for systems reflecting business needs. This is one of the reasons, why we have chosen BORM methodology (Business Object Relation Modeling) (Knott et al. 2003) – an object-oriented methodology for information system design and development based on the business process modeling. BORM itself focuses to modules from reusability point of view and its focus is also to business process modeling. These aspects are similar to SOA principles. But the way of information system design is pure object-oriented. More about this methodology can be found in the following section.

## 2 BORM methodology

BORM – Business Object Relation Modeling was originally developed in 1993 and was intended to provide seamless support for building object-oriented software systems based on pure object-oriented languages and environments such as Smalltalk and object databases. Subsequently, it has been realized that this method has also significant potential in business process modeling and other related business and user requirement issues.

Business Object Relation Modeling (BORM) is an approach to both process modeling and the subsequent development of information systems. It provides a methodology that facilitates the description of how real business systems evolve, change and behave. It has been used successfully for developing a number of information systems in various areas of business activities. The BORM approach is based on each object having three independent attributes called dimensions. These are data, behavior and history (a composition of states and transitions, i.e. the object lifecycle).

BORM is fundamentally an object-oriented development methodology, but differs from other such methodologies. In BORM, the extent of knowledge required to understand an object and use it effectively in the design process evolves

throughout the development process in a clear, precise and consistent manner. Initially, objects are defined as business objects, where only knowledge of their activities, relationships, and intercommunications is required (Satzinger and Orvik 1996). Business objects during the design process are changed via a set of clearly defined and consistent techniques into conceptual objects. During the implementation phase conceptual objects evolve in a similar structured and controlled manner into software objects. Thus at each stage of the development process, BORM requires some degree of knowledge about the object to proceed further.

Contrary to some other object-oriented methodologies, which start with a set of initial objects without providing any method for discovering them, the BORM development methodology 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 main technique used here is BORM modified Object Behavioral Analysis (BOBA).

BORM, similarly to other object-oriented development methodologies is based on the spiral model for the development life cycle. One loop of the object-oriented spiral model contains stages of strategic analysis, initial analysis (both work with business objects), advance analysis, initial design (conceptual objects), advanced design, implementation and testing (software objects):

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

## 3 Extending the BORM considering SOA

Our goal was to examine if the object models created within BORM development can be utilized to support the design of service-oriented systems and to investigate under which conditions this procedure will work.

For the purpose of practical investigation of our approach an experimental project from the domain of self-service DVD rental store was established. The project consisted of three phases. In the first step design via original BORM was per-formed. In the second step the design respecting SOA principles was made. And in the last step consolidation of these two designs was completed.

BORM design was performed according to its six stages. The SOA design was built considering SOA principles and Business Process Execution Language (BPEL) (Juric et al. 2006). BPEL models were used for visualization of service orchestration. These models provide complete view to service architecture.

Comparing BORM and BPEL processes, a similarity in identified activities was found. BORM process models are comparable with UML activity and state diagrams whereas they combine both state and activity aspects of processes in a single diagram (see also Polášek et al. 2006). Besides vertical relations they also capture horizontal ones, which describe communication among processes of various participants. The similarity between BORM and BPEL processes lies in identified services and in process map where the services were orchestrated in related way – in terms of nesting, cycling, branching etc.

The differences between models are caused due to their special purpose. BORM models are created in the first stages of system development in order to describe problem area whilst BPEL models have to be designed in the latter phases based on well-known user needs and expectations. But BORM's process models are a good background for building SOA and BPEL models eventually – if they are desired. Speaking more generally, when comparing BORM and SOA designs there was found out that support of services identification as functional elements (for SOA) together with their forming into cooperative unit is missing in BORM. The cooperative unit of formed and also orchestrated services is called a "process layer" in this paper. It is necessary to realize that there is a difference between the process layer on one hand and a business layer, business process and process diagram/model on the other. They are considered as two different models.

The gap between BORM and SOA should be filled. BORM methodology does not provide all the necessary information. Some additional data must be added for recognition of relevant activities as services. Moreover, their classification and grouping should be performed. Into the original BORM's six phases the additional activities were added taking into account the nature of each phase as well as its overall purpose in the methodology.

**First and second phase.** The aim of the first two phases of BORM is to describe a problem area. The requirements of the future system are defined along with process diagrams. These phases include all the necessary activities – also for SOA. Hence an upgrading of these phases is not required. In relation to the experimental project, project objectives were defined, together with sets of required system functions and system scenarios. And also the business processes for operators (customer administration + movie administration) and customers (movie rent and return, movie reservation and canceling of reservation) were recognized.

**Third phase.** In the third phase, process diagrams are processed and information for an initial software specification is extracted. The following activities were added to this phase in order to support SOA: initial service identification (based on process models), their classification – grouping according to their logic and decomposition of services, if it is appropriate. In this phase of the system design, object diagram and class diagram with relationships between objects, resp. classes has been created together with its extension by objects' dynamic properties. For supporting SOA approach initial identification and classification of services based

on business processes have been executed. Example of business services for customer is shown in Table 1.

**Table 1** Initial list of customer business services

| *Participant* | *Business services* |
|---|---|
| Customer | person's identification and authentication<br>person's log off<br>administration of borrowings<br>creation of list of rented movies<br>creation of list of reserved movies<br>finding a movie<br>movie reservation<br>canceling of movie reservation<br>stock out of movies reserved on current branch store (in case of picking up the movies)<br>receiving of movie (in case of returning borrowed movie) |

**Fourth phase.** Design in the fourth phase proceeds closer to software implementation. The relationships among objects are improved and transformed. Design patterns are applied and reusable components identified. From SOA point of view the existing services are identified and the input and output parameters are specified.

Business services are considered as SOA operations and bundled to SOA services according to theirs functionality. It is recommended in this phase that analysts consult together with developers identified services from implementation point of view. This is important for the modifications due to effective implementation (it should be appropriate for decomposition, consolidation, regrouping of services etc.). Outputs of this phase are object diagrams enriched by design patterns and a list of regrouped SOA operations bundled in services. The services have been grouped according to the area they deal with. An example of customer service with its operations is shown in Table 2.

**Table 2** List of SOA services and theirs operations for customer service with input and output parameters

| *Service* | *Operation* | *Input parameters* | *Output parameters* |
|---|---|---|---|
| Customer | person's identification and authentication | customer id, branch store id | 1 – successful<br>0 – unsuccessful |
| | person's logging off | customer id | 1 – logging off successful<br>0 – logging off unsuccessful |

| *Service* | *Operation* | *Input parameters* | *Output parameters* |
|---|---|---|---|
| | registration of new customer | name, address | 1 – registration successful<br>0 – registration unsuccessful |
| | canceling of customer registration | customer id | 1 – canceling successful<br>0 – canceling unsuccessful |
| | person's identification via identification card | card id | customer id and name |
| | modification of customer data | customer id, new data | 1 – data modification successful<br>0 – data modification unsuccessful |
| | listing of customers | - | list of customers with their data |

**Fifth phase.** In the fifth phase a transformation of conceptual objects closer to concrete implementation environment is made. Besides adjusting the class and object diagrams also process models of services are prepared in order to form a process layer of implementation. Design of the process layer where services and relations among them are captured, is made for later implementation purposes. It brings an outline for implementation of previously orchestrated services. The process diagrams from the second phase are helpful in this phase and the process layer design can be based on them.

The process layer can be designed using either workflow patterns, business process modeling notation (BPMN) or eventually BPEL. BPEL should be used in this phase to create a high level design of process layer in case of later implementation of process layer in this language.

In the experimental project the diagrams of software objects capturing a server side with relation database have been created. Then a process layer has been designed (see Fig. 1), NetBeans BPEL notation has been used.

**Sixth phase.** In the sixth – final phase, the physical implementation of the system including implementation of identified services using web service standards was performed. The process layer, which is used for services interconnection is also created.

A prototype of a self-service DVD rental store has been implemented in Java with MySQL. The Java application consists of the web service layer, in which single SOA operations have been implemented, and of the process layer, which is bundled with user interface implemented with Java Server Faces and of MySQL

database. In case of the process layer implementation using BPEL, we will get really flexible and scalable system as SOA proclaims.

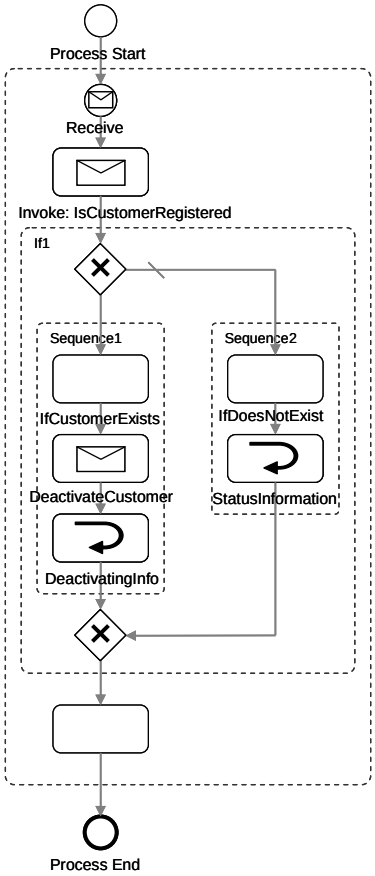


**Fig. 1** Design of a process for customer deactivation

Our experiment has demonstrated that if the development of information systems according to BORM methodology is performed thoroughly, we can take advantage of the created models of business processes to support the system design respecting service oriented approach. Moreover, not only the models of business processes are profitable. As we have shown above, also extended diagrams of objects and classes together with their relationships and dynamic properties from the BORM development can be with benefit utilized in the process of development of service-oriented systems.

An overall life cycle of extended BORM methodology is shown in Fig. 2 (BORM methodology is depicted with inspiration from (Molhanec 2008)).

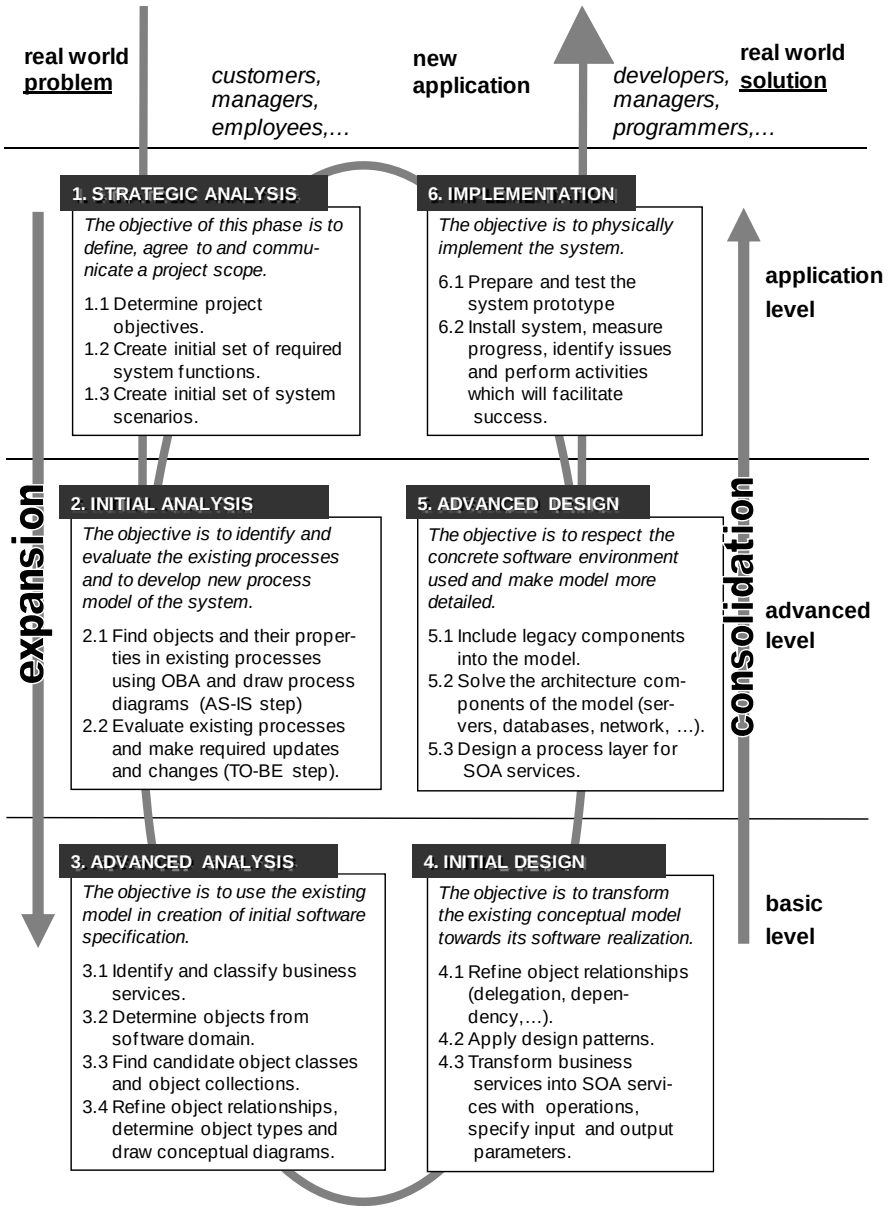


**Fig. 2** Life cycle of extended BORM methodology

## 4 Conclusion

This paper investigates the possibilities of an extension of BORM methodology by the SOA approach. The development following all the steps of the original BORM sequence, augmented by activities allowing SOA principles to be a part of the design, has shown to be viable – thus making a step towards new methodology of information system design and development.

We found that in case of good-quality design by BORM it is possible to use the process models for initial service identification and for process layer implementation according to service oriented architecture. It is also possible to make use of other BORM diagrams from initial steps of system analysis and design. As a result of BORM extension, an adapted methodology with extended phases of the life cycle was created. This methodology is based on object-oriented approach taking into account requirements needed for service oriented principles. The extension has effect on the last four out of six phases of the original BORM methodology. First two phases describe problem area deep enough also for SOA. Their extension was not necessary. The new upgraded methodology allows smooth design of information systems facilitating to build the systems, which are service-oriented. An implementation of system prototype via extended BORM methodology served for evaluation of proposed extensions.

By finding the way of extending the original methodology by SOA we achieve a methodology that enables identification and implementation of an information system process layer. The realization of the layer can be done in several ways. If we choose BPEL, these changes can be performed on BPEL process model level, utilizing advantages of BPEL engine. Thus we have the possibility to make fast and flexible modifications of information systems according to actual needs, and assuming that flexibility belongs to the most desirable attributes of information systems today, perhaps we could be a step further in the methodologies of their development.

**Acknowledgements** This work was partially supported by the Slovak Research and Development Agency under the contracts No. APVV-0391-06 and No. APVT-20-007104.

## References

1. Bell M.(2008). *Service-Oriented Modeling (SOA): Service Analysis, Design, and Architecture.* John Wiley & Sons, New Jersey.
2. Erl, T. (2008). *SOA: Principles of service design.* Prentice-Hall, Boston
3. Juric M., Mathew B., Poornachandra S. (2006). *Business Process Execution Language for Web Services,* Packt Publishing, Birmingham, UK
4. Knott R. P., Merunka V., Polak J. (2003). The BORM methodology: a third-generation fully object-oriented methodology. *Knowledge Based Systems,* 16 (2): 77–89.

5. Molhanec, M. (2004). An overview of Information Modelling for Manufacturing Systems In: 27th Int. Spring Seminar on Electronics Technology ISSE 2004. Sofia: IEEE, 2004, vol. 1, pp. 36-41.
6. Polášek,I., Chudá, D., Kristová, G. (2006). System dynamics modeling in the new version of UML 2.0 language. In: *System Integration 2006*, Žilina, University of Žilina, pp.311-317.
7. Satzinger J. W., Orvik T.U. (1996). *The Object-Oriented Approach - Concepts, Modeling and System Development*, Boyd&Fraser.