SEM 2013

1st International Workshop in Software Evolution and Modernization

Ricardo Pérez-Castillo (Ed.)

Proceedings of SEM 2013

1st International Workshop in Software Evolution and Modernization
In conjunction with the 8th International Conference on Evaluation of Novel Software Approaches to Software Engineering - ENASE 2013

ESEO, Angers Loire Valley, France

Ricardo Pérez-Castillo (Ed.)

Software Evolution and Modernization

Proceedings of the
1st International Workshop in
Software Evolution and Modernization
SEM 2013

In conjunction with ENASE 2013
Angers, France, July 2013

SCITEPRESS
Portugal

Volume Editor

Ricardo Pérez-Castillo
Instituto de Tecnologías y Sistemas de Información (ITSI),
University of Castilla-La Mancha
Spain

1st International Workshop in Software
Evolution and Modernization
Angers, France, July 2013

Printed in Portugal

ISBN: 978-989-8565-66-2
Depósito Legal: 360518/13

Foreword

Overall Objectives and Mission

Welcome to SEM 2013, the First International Workshop on Software Evolution and Modernization, held July 5 in ESEO, Angers Loire Valley, France, and co-located with the 8th International Conference on Evaluation of Novel Approaches to Software Engineering Software (ENASE 2013).

The scope of this workshop focus mainly on legacy information systems, which can be a serious headache for companies because, on the one hand, such systems cannot be thrown away since they store a lot of valuable business knowledge over time, and on the other hand, they cannot be maintained easily at an acceptable cost. For many years, reengineering has been a solution to this problem because it facilitates the reuse of the software artifacts and knowledge embedded in the system. However, reengineering often fails due to the fact that it carries out non-standardized and ad hoc methods. This workshop attempts to explore and provides novel approaches in software modernization and evolution taking into account new trends in this field.

The overall objective of this workshop is to provide a mean for researchers, designers, and practitioners to share experiences, new trends and future challenges about Software Evolution and Modernization. This will also showcase the heterogeneous technology solutions implemented in different parts of the world in the realm of Software Evolution and Modernization.

The target audience of this workshop is composed of professionals and researchers working in the field of information and knowledge management in various disciplines, e.g. government departments, information and communication sciences, e-Government practitioners and designers, university departments, computer science, and information technology. Moreover, the workshop provides insights and support to executives concerned with the management of expertise, knowledge, information, and organizational development in different types of software maintenance communities and environments.

Topics Covered

In total, the first edition of this ENASE workshop consists of 9 full papers as well as 4 short papers, with co-authors of eight different countries. Each paper received three reviews by members of the program committee. After we received all reviews, an online discussion process allowed variance among reviews to be understood, and in many cases, consensus to emerge.

The main topics included in the accepted papers were: Data and Process Mining; Model-Driven Engineering; Software Architecture Recovery Service as well as Engineering and SOA; Program Transformation and Refactoring; and several contributions about Software Testing.

Papers were grouped into four different sessions. The first session was devoted to Model-Driven Maintenance, in which three papers were included. Traceability vs. Quality. Which is the opportunity cost for ATL model transformations? from Santiago et al. presents an empirical study to assess the impact on quality of the enrichment of ATL model transformations. An ADM-based Method for migrating CMS-based Web applications: extracting ASTM models from PHP code from Trias et al. defines an ADM-based method for migrating CMS-based Web applications. Challenges of Business Process Model Improvement after Reverse Engineering from Fernandez-Ropero et al. describes all detected challenges that should be addressed for improving quality of business processes, specially retrieved by reverse engineering.

The second session, which was framed in Software Testing, contains three full papers. From Functional Test Scripts to Performance Test Scripts for Web Systems from Toledo et al. presents a tool that we developed for industrial usage to automatically generate performance tests scripts from automated functional tests. Towards Effective and Efficient High Order Mutation from Polo and Reales proposes and evaluates a novel type of mutation testing, mutants-integration mutation that improves the effectiveness of High Order Mutation. Impact-Driven Regression Test Selection for Mainframes from Dharmapurikar et al. is aimed at reducing the required regression testing and its costs associated with the system.

The third session consists of four short papers: Enhance your Model-driven Modernization Process with Agile Practices from Krasteva et al.; Realization of Agile Enterprises: A Meet-in-the-Middle MDA Approach for Service-Oriented Business Processes from Baghdadi et al.; Upgrading Unupgradable Legacy Middleware Systems: Misconceptions, Challenges, and a Roadmap from Jrad et al.; and ETL-Text: Extract-Transform-Load Processes for Textual Data Warehousing from Rachid Aknouche.

Finally, the miscellaneous session contains three papers about Architecture Reconstruction. Investigating the Applicability of Lehman's Laws of Software Evolution using Metrics: An Empirical Study on Open Source Software from Badri and Drouin aims at investigating empirically the applicability of Lehman's laws of software evolution using software metrics. Software Architecture Reconstruction Through Clustering: Finding the Right Similarity Factors from Ioana Sora investigates the importance of taking into account different factors for the similarity metric in clustering algorithms. Extracting services from legacy for service-oriented business processes: challenges, issues, and solutions from Baghdadi and Pérez-Castillo advocates using a service-oriented business modeling by stressing out role of SOA and Web services and achieve a higher agility.

July 2013,

Ricardo Pérez-Castillo
Instituto de Tecnologías y Sistemas de Información (ITSI),
University of Castilla-La Mancha, Spain

Workshop Chair

Ricardo Pérez-Castillo
Instituto de Tecnologías y Sistemas de Información (ITSI),
University of Castilla-La Mancha
Spain

Program Committee

Youcef Baghdadi, Sultan Qaboos University, Oman
Fabián Baptista, Abstracta, Uruguay
Danilo Caivano, University of Bari, Italy
María Fernández-Ropero, University of Castilla-La Mancha, Spain
Chiara Di Francescomarino, Fondazione Bruno Kessler, Italy
Jorge J. Gomez-Sanz, Universidad Complutense de Madrid, Spain
Ignacio García Rodríguez De Guzmán, University of Castilla-la Mancha, Spain
Ravi Khadka, Utrecht University, The Netherlands
Alessandro Marchetto, Fondazione Bruno Kessler, Italy
Pedro Reales Mateo, Alarcos Group, Dept. of Information Systems and Technologies, University of Castilla-La Mancha, Spain
Ricardo Pérez-Castillo, Instituto de Tecnologías y Sistemas de Información (ITSI), University of Castilla-La Mancha, Spain
Federico Toledo Rodríguez, Abstracta, Spain

Table of Contents

Full Papers

Short Papers

Realization of Agile Enterprises: A Meet-in-the-Middle MDA Approach for Service-Oriented Business Processes

Youcef Baghdadi[1], Naoufel Kraiem[1,2], Yassine Jamoussi[1,2] and Ricardo Pérez-Castillo[3]

[1]Department of Computer Science, Sultan Qaboos University
P.O. Box 36 – PC 123, Al-Khod, Oman
[2]RIADI Lab, University of Manouba, Campus Universitaire de Manouba, Tunis, Tunisia
[3]Alarcos Research Group, University of Castilla-La Mancha,
P. de la Universidad, 4 13071, Ciudad Real, Spain
{ybaghdadi, naoufel, jamoussi}@squ.edu.om,
Ricardo.PdelCastillo@uclm.es

Abstract. Enterprise agility is not possible if Business Processes (BPs) are rigid and cannot respond to the environmental changes. One of the solutions is to make service as part of the requirements and solution, by using services as the main building blocks at different levels of abstraction. In this approach, we model BPs by using specialized services having separated concerns. The enterprises business objects, as implemented in the information system, provide these services. This approach requires both reverse and forward engineering. The former presents the existing information systems as a set of common, consistent, sharable BOs. Whereas, the latter uses MDA to take profit of the rapid transformation of the BP models into executable by using standards such as BOs, web services and BPEL.

1 Introduction

To become agile, enterprises have to adapt their BPs in response to different Business Events (BEs) such as different customer demand, new acquisition, or merging. The achievement of this agility is difficult as: (i) BP modeling considers that all the instances of the same BP run the same way, (ii) the architecture of the existing Information System (IS) is siloed, which may not fully support the BPs by providing them with the necessary data.

One of the solutions is to reengineer the BPs, by using a service-oriented business modeling that stresses out the roles of SOA and web services, and the LIS. In this modeling, a BP is first defined as a dynamic service that responds to BEs of different natures. It specifies a state that reflects the control flow and data flow, where changes in BEs are solely reflected in the state. The BP adapts to the state changes, which allows a dynamic realization. This service is itself realized by enterprise service units that play the role of service providers or units [1]. Then, it is modeled by using specialized services having separated concerns. These are: (i) controller service, (ii) state service, and (iii) worker services, where the controller service uses the state service to select the worker services to invoke [2]. The worker services are themselves provided by shared, discoverable, reusable enterprise assets known as Business Objects (BOs). In this modeling, the IS is constituted of a set of BOs, where a BO is “a representation of a thing active in the business domain, including at least its business name and definition, attributes, behavior, relationships, and constraints” (OMG). BOs model the enterprise service units that encapsulate activities and data; and provide them as services to all BPs.

This solution requires an integrated view of all the reengineering phases [3], i.e., reverse engineering, restructuring and forward engineering. The reverse engineering first transforms the IS as a collection of BOs and presents the BPs with service-oriented perspective. After that, during the restructuring stage the BP models are modified and new business requirements are met to achieve agility. Finally, the forward engineering phase (which is addressed by this paper) is based on MDA [4] to take profit of the rapid transformation of the BP model into executable standards such as web services, BPEL, and WS-CDL. According to the MDA principles, three models at three different levels have to be considered:

– At the Computational Independent Model (CIM) level, the modeler specifies the multi-valued state to model the BP.
– At the Platform Independent Model (PIM) level, the BP model is transformed by using rules.
– At the Platform Specific Model (PSM) level, the BP model is transformed into an executable process such BPEL

This approach would have practical implications in terms of (i) compliance with SOA for better integration and composition of BPs that responds to changing events, and (ii) rapid modeling, enacting, and execution of BPs.

The remainder of this paper is organized as follows: Section 2 provides some related work. Section 3 details the concepts of the BP modeling. Section 4 presents the approach. Finally, a conclusion section provides future work.

2 Related Work

This work concerns with three perspectives: (1) modeling BPs as services, (2) reverse engineer supporting ISs with respect to SOA, and (3) using MDA to forward engineer BPs. From service-oriented BP modeling, the academics have raised the importance of service-oriented as one of the top three issues to deal within BPs [5]. In [6], the authors introduced service-oriented requirements engineering as a discipline aiming at a better and more systematic handling and alignment of SOA and BPM. In [7], the author proposed a business model for B2B integration through web services. The authors in [8] proposed an approach for designing BPs in service-oriented way, where a service composition process organizes the discovery, selection and assembly of business services to build BPs tailored to business designer's requirements. In [9], the author investigated how to extend Event-Process Chain (EPC) to come up with a new modeling language for service-oriented BP management.

From a general perspective of modernizing ISs and BPs, different approaches have been proposed [10]. In [11], the authors developed Marble to reengineer BPs within a BP archeology. This approach makes it possible to obtain BP models from source code and other software artifacts by reverse engineering. Efforts that concern with modernization for SOA have been reported in [12]. In [13], the author proposed to reverse engineer relational databases into services. In [14] the authors have defined a technique to wrap legacy applications for reuse in a SOA.

From the perspective of mapping enterprise services into a given service-oriented business modeling, there is a lack of approaches, though many authors such [15], [16], and [17] have attempted to identify and classify services. For instance, the authors in [18] propose to modernize legacy software system by using web services as the main building blocks of the software reengineering.

From MDA perspective, only top-down approaches such as SoaML [19] have been investigated to allow identification and specification of services. In [20], the authors propose an MDA-based transformation technique for service composition. In [21] [21], the authors propose a framework that adopts a holistic perspective on interoperability in order to analyze and understand the business needs and the technical requirements, and a multidisciplinary and model-driven solution approach to solving the interoperability problems.

The drawback of these approaches is that they do not explicitly consider the reengineering of the current LIS to provide the services that the agile enterprise requires, neither have they used an MDA approach for service-oriented BP modeling.

3 Service-Oriented Business Process Modeling

A BP has a value expressed in terms of quality of services to customers. That is, a BP is a description of a service provided to customers upon their demands (of different natures). A set of coordinated activities realize the service. The IS supports these activities by providing them with data they consume.

A BP modeling captures the relevant properties with respect to the above-mentioned definition of BP. In our service-oriented modeling approach, specialized services are used as constructs to represent the different relevant elements of BPs.

A BP has a set of state values (including initial and final state values) that the BP modeler sets and changes over time when business requirements change. A state value reflects the execution BP.

A ***service*** may be defined from business and technology perspectives [22]. In our modeling, we use web services and data services.

Our service-oriented approach for BP modeling emphasizes the separation of concerns that differentiate the activities of control and execution. Similarly, the data packaged into BOs are separated from the state of the BP. Therefore, we specialize a service into controller service, state service, and worker service.

The ***Controller Service*** (CS) is the central element of our modeling. It oversees a BP execution through its state. The CS deals only with the control and coordination of the BP. It invokes a State Service (SS) to retrieve the state of the BP; and accordingly invokes the respective Worker Service (WK) and updates the BP state when any of the WS terminates its job. The CS is invoked by an Initiator Web Service (IWS).

The ***Initiator Web Service*** (IWS) deals only with the initialization and starting of an instance of the BP.

The ***State Service*** (SS) is a data service that represents the structure of the state of a BP. It is used by the CS to retrieve the state of the BP.

The ***Worker Services*** (WKs) add value to a BP towards the achievement of its goal. WKs are provided by BOs. The CS according to the BP state invokes them. Inversely, WKs report the outcome of their execution to the CS.

WKs may use ***Data Services*** (DS) if necessary to retrieve simple or integrate data from BOs. DSs retrieve or integrate the requested data from the BOs.

Business Object is "a representation of a thing active in the business domain, including its business name and definition, attributes, behavior, relationships, and constraints" (OMG). To represent BOs in a consistent way regardless of the needs of BP modeling, we insist on the monolithic representation of the activities and data of these BOs. Unfortunately, this representation is not happening nowadays as different representations (or images) of the same BO are developed depending on the needs of each organizational body, which leads to a limited view and inconsistency of BO across the whole organization.

There are four types of relationships between these concepts: association, specialization, realization, and use.

- ***Association relationships*** indicate how the elements of a BP environment are associated with each other. For instance, a BP is associated with an event, a set of BOs, a set of states, including an initial state, and a final state.
- ***Specialization relationships*** indicate the specific roles of some elements of the model. For instance, a service may be a CS, a SS, or a WK.
- ***Use relationships*** show that some services use the capabilities of others. For instance, the CS uses both WKs and SS. The latter provides it with the state of the BP, whereas the former perform the required activity. The IWS uses the CS.
- ***Realization relationships*** show that WKs are realized in the IS by the BOs, whereas the SS is realized by a specific data structure representing the state values.

4 MDA Approach for Service-Oriented Business Process Modeling

The MDA approach uses models at three levels and the mapping between models. These are Computational Independent Models (CIM), Platform Independent Models (PIM) and Platform Specific Models (PSM). Using MDA to forward engineer the BPs mainly consists in modeling BPs at all levels by services. Table 1 summarizes the content of each MDA level (Column 2) along with the involved stakeholders (Colum 3), the techniques they use (Column 4), and their supporting information (Column 5). Figure 1 sketches out the transformation approach. Since we are using standards whenever applicable, the CIM uses an activity diagram instead of an ordered graph of (business services), and the PSM uses the BPEL. The PIM contains a service-oriented BP modeling.

Table 1. The process.

DA	*Content*	*Stakeholder*	*Techniques*	*Sources*
CIM	BE Value (context, goal) Strategies Initial and Final States Activity Diagrams	BP Modeler BP Developer	BP Management Business Modeling Reverse Engineering	IS
PIM	Option 1: Mapping activity diagrams into composite service (CS) Option 2: Mapping activities into Worker Services	BP Developer	Forward Engineering	BOs Registry
PSM	Option 1: Mapping CS into BPEL Option 2: Developing CS Option 3: Mapping Worker Services into Partner Services	BP Developer	Forward Engineering	Implemented BOs

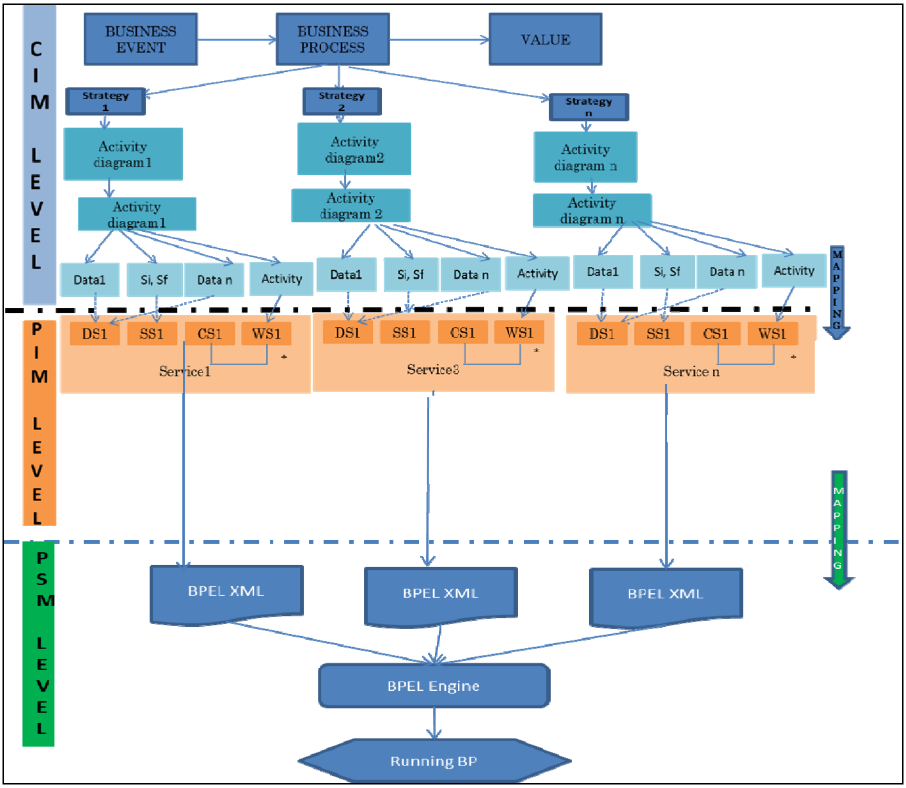

Fig. 1. Transformation approach.

4.1 MDA Levels

At CIM Level, the modeler determines:

- The Business Event (BE) that needs the execution of an artifact (BP service) in order to get a response
- Value (to consumers) : context aware value (context, goal)
- The enterprise assets (e.g. IS)
- A multi-valued state that is used to control the BP, by defining each value of the state in terms of activities
- State values that corresponds to the different exceptions
- Different strategies, where each strategy deals with a distinct type of BE

At PIM Level

- BOs playing the role of service providers
- BO interfaces, where an interface present all the activities and data a BO may provide
- WKs that wrap the BO activities into web services
- DSs that wrap the BO data into data services
- SS encapsulates the different values of the state and provides them as services. It is used by the CS
- CS that controls the flow of execution of the WKs

At PSM Level

Since our aim to use standards as much as possible, each CS would be mapped into BPEL. However, any other executable language could be used instead of BPEL

4.2 Model Transformations

CIM to PIM

A CIM is the view of a system from a computation independent viewpoint. CIM describes BPs without any reference to the solution. Stakeholders use this description to build the architecture, which requires CIM must to be expressed in understandable language.

UML is the most referred modeling language. Its last version, UML2, is widely adopted for modeling not only the application structure, but also its behavior, and architecture [23]. Therefore, we recommend expressing the BP with UML diagram. The activity diagram can describe the BP clearly (we can identify easily the BOs and DSs) with a data flow and object's state (useful to determinate the SSs).

The final goal of this approach is to develop a set of services. Each service expresses a part of the BP.

In the PIM phase, we will transform data flow to a data service and the state object to a state service. Using an analyzing data we can identify the controllers needed by the system (each data transformation is an execution method or function). With the flow of data we can also detect the relationship between services. For example, if service *j* needs data providing from service *k* so we can say that service *j* "uses" service k.

PIM to PSM

In the PSM, the choice of the service type (web service, BPEL, etc) will be explicitly defined and modulated. The transformation of PIM to PSM depends on this choice. Every element of PIM (DS, SS, or WK) will be transformed to conform to this architectural choice. In Figure 1, the service type is BPEL. The PIM artifacts are WKs, SS, DSs and CS.

A set of transformation operation was executed (mapping, BPEL XML, BPEL engine) to build a BP PSM. The PIM provides reuse (the goal of MDA approach). Once we want to change service type, we have just to retransform artifacts.

5 Conclusions

We have defined a service-oriented business process modeling with respect to a certain level of maturity, where services are part of the requirements and the solution. This would smooth the transformation from higher abstract level to lower ones. That is elements of business process environment are modeled as services, including controller service, state service and worker services.

Then, we have used MDA for forward engineering to reuse existing standards, namely activity diagram, at CIM level, and BPEL, at PSM level.

Then, we have sketched out the transformation process from CIM to PIM and from PIM to PSM.

Although, we have presented approaches and processes for transformation techniques having a real impact on the way BPs should responsive to the changes in the business requirements by using web service-based SOA, this work has limitation. Therefore, we would consider that this work has presented the service-oriented business process as rather a roadmap towards research issues and questions related to transformation techniques than a definitive solution. We need to develop tools and dig deeper in the different transformation techniques.

We need to develop tools and dig deeper in the different transformation techniques

References

1. Cummins, F. A.: Building the Agile Enterprise: With SOA, BPM and MBM, Morgan Kaufmann, 2010.
2. Baghdadi, Y.: A methodology for web services–based SOA realisation, Int. Journal of Business Information Systems, 10(3), pp. 264-297, 2012.
3. Chikofsky, E. J. and Cross, J. H.: Reverse engineering and design recovery: a taxonomy, IEEE Software, Vol. 7, no. 1, pp. 13–17, Jan. 1990.
4. Kleppe, W., Wramer, A. Bast, J.: MDA Explained, 2003.
5. Indulska, M., Recker, J., Rosemann, M. and Green, P.: Business Process Modeling: Current Issues and Future Challenges, Advanced IS Engineering, LNCS, Vol. 5565, 2009, pp 501-514, 2009.
6. Adam, S., Riegel, N. and Doerr, J.: From business processes to software services and vice versa—an improved transition through service-oriented requirements engineering, Journal of Software Evolution, Vol. 24, No. 3, pp. 237–258, 2012.

7. Baghdadi, Y.: A business model for B2B integration through Web services, in Proc. of IEEE Int. Conference on e-Commerce Technology, 2004. CEC 2004, pp. 187–194, 2004.
8. Cauvet, C. and Guezilian, J.: Business Process Modeling : a Service-Oriented Approach, in Proc. of Hawaii Int. Conference on System Sciences, 41st Annual pp. 1–8, 2008.
9. Stein, S.: Modelling Method Extension for Service-Oriented Business Process Management," Thesis, 2008.
10. Rahgozar, M. and Oroumchian, F: An effective strategy for legacy systems evolution, Journal of Software Maintenance and Evolution: Research and Practice, Vol. 15, no. 5, pp. 325–344, Sep. 2003.
11. Pérez-Castillo, R., de Guzmán, I. G.-R. and Piattini, M.: Business process archeology using MARBLE, Information and Software Technology, Vol. 53, no. 10, pp. 1023–1044, 2011.
12. Khadka, R., Saeidi, A., Idu, A. Hage, J and Jansen, S.: Legacy to SOA Evolution : Evolution : A Systematic Literature Review, Technical Report UU-CS-2012-006, 2012.
13. Baghdadi, Y.: Reverse engineering relational databases to identify and specify basic Web services with respect to service oriented computing, Information Systems Frontiers, Vol. 8, no. 5, pp. 395–410, Nov. 2006.
14. Sneed, H. M., Schedl, S and. Sneed, S. H.: Linking legacy services to the business process model, in IEEE 6th International Workshop on the Maintenance and Evolution of Service-Oriented and Cloud-Based Systems (MESOCA), pp. 17–26, 2012.
15. Al-Rawahi, N. and Baghdadi, Y.: Approaches to identify and develop Web services as instance of SOA architecture, In Proc. of ICSSSM '05. Int. Conference on Services Systems and Services Management, 2005, pp. 579–584, 2005.
16. Cohen, S.: Ontology and taxonomy of services in a service-oriented architecture, The Architecture Journal, Vol. 11, pp. 30-35, 2007.
17. Baghdadi, Y. and Al-Rawahi, N.: An architecture and a method for Web services design: Towards the realization of service-oriented computing', Int. Journal of Web and Grid Services (IJWGS: InderScience), 2(2): pp 119-147, 2006.
18. Chung, S., Young, P. and Nelson, J: Service-oriented software reengineering: Bertie3 as web services, Web Services, 2005. ICWS …, 2005.
19. Poler, R., Doumeingts, G., Katzy, B. and Chalmeta, R.: Enterprise Interoperability V. London: Springer London, 2012.
20. Khadka, R., Sapkota, B. and Pires, L: Model-driven development of service compositions for enterprise interoperability, Enterprise Interoperability, LNCS, Vol. 76, 2011, pp 177-190, 2011.
21. Berre, A., Elvesæter, B. and Figay, N.: The ATHENA interoperability framework, Enterprise Interoperability II, pp 569-580, 2007.
22. Chesbrough, H. and Spohrer, J.: A research manifesto for services science, Communications of the ACM, vol. 49, no. 7, p. 35, Jul. 2006.
23. Song, I.Y. et al.: Advances in Conceptual Modeling – Challenges and Opportunities, Vol. 5232. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008.