Model Transformation Recommendations for Service-Oriented Architectures

Dmitri Valeri Panfilenko[1], Katsiaryna Hrom[2], Brian Elvesæter[3]and Einar Landre[4]
[1]*DFKI GmbH, Saarbrücken, Germany*
[2]*T-Systems International GmbH, Saarbrücken, Germany*
[3]*SINTEF, Oslo, Norway*
[4]*Statoil, Stavanger, Norway*

Keywords: SOA, MDA, Model Transformations, Recommendations, CIM, PIM, M2M, BPMN, SoaML.

Abstract: Services for service-oriented architectures can be modelled in different ways, including well-known existing OMG standard SoaML and an IBM methodology SOMA. Involving domain expert stakeholders in the system specification and development process plays an important role and is often inevitably combined with model transformations between different levels of abstraction. Recommendations for those supporting users during the modelling process along the chosen methodology can aid the development performance and thus reduce model transformation efforts. This paper shows how bidirectional model transformations between OMG MDA's CIM and PIM levels can be enhanced through recommendations and which obstacles on the way to a comprehensive framework for model-driven development are still to overcome.

1 INTRODUCTION

Nowadays it is impossible to think away from the software systems in the enterprise workaday life because they are facilitating a supportive role for the enterprise management in the market competition. The need for software solutions in the business practice faces increasing complexity of these solutions and permanently rising requirements for performance, reliability and shorter technology cycles. In addition, the changing requirements and pressure for cost reduction count as well. The new concepts and techniques for software engineering as an actual problem are not to be undervalued.

The core principles of SOA – loose coupling, atomicity and consistency (Erl, 2007) – are aspects present in every system developed in this way, but transferring the information from the higher levels of abstraction as domain knowledge to the technical system specification has always been an issue researched upon (Kleppe, 2003); (Petrasch, 2006).

An MDA approach provides a basic scheme of separation of concerns of domain experts and technical specialists, giving them an opportunity of working together and not having to get much involved with concerns of other domains or abstraction levels (research projects SHAPE: www.shape-project.eu, REMICS: www.remics.eu).

The problem of transferring the information between different abstraction levels is the background of the present paper. We try to amend the existing approaches for the model transformations by engineering recommendations for the end-users facilitating more automation in the information propagation.

The paper flow proceeds as follows: section 2 describes the existing work in the areas of model transformation and recommendation engineering. Section 3 describes model transformation application for engineering of service-oriented systems. Section 4 presents a case study elaborated in SHAPE project and evaluation of it. We discuss strengths and weaknesses as well as implications from the existing approaches to the model transformation and recommendation engineering in section 5. The section 6 provides conclusions and sketches the future work plans in these areas.

2 RELATED WORK

The implementation of the MDA approach on the CIM and PIM level is a field of interest for industry and science. According to the literature survey of

Panfilenko D., Hrom K., Elvesæter B. and Landre E..
Model Transformation Recommendations for Service-Oriented Architectures.
DOI: 10.5220/0004445602480256
In *Proceedings of the 15th International Conference on Enterprise Information Systems* (ICEIS-2013), pages 248-256
ISBN: 978-989-8565-60-0

existent MDA and MDA-based implementations the most of them ignore the CIM level and therefore the related model transformations (ATLAS, 2005); (Jeary, 2008); (Petrasch, 2006). At the same time CIMs and PIMs present business and analysis level of software systems correspondingly and the models of this level have a big impact on the software development effort and the quality of software.

Quite recently many authors have mentioned that automatic transformations from PIM to CIM are not possible (Kahl, 2005); (Kleppe, 2003); (Koch, 2006). One of the last published practical researches about CIM-to-PIM transformations showed that the examined transformations could not be completely automated "because it is needed both to add information concerning the context and to make decisions" (Lemrabet, 2010).

In the last few years there have been done many research attempts aiming to apply MDA (or MDD) paradigms to business processes and services. One of the most similar works to our research is that one from (Delgado, 2010) where MINERVA's tool support for service oriented development from business processes is presented. This tool support includes QVT automated model transformations from BPMN to SoaML. De Castro et al. describe the transformations between CIM and PIM models for model-driven process (De Castro, 2011). In their previous work (Rodríguez, 2007); (Rodríguez, 2008) have proposed a CIM to PIM transformation composed of QVT rules. (Touzi, 2009) designed collaborative SOA according to MDA principles. As a part of work the BPMN-to-UML transformations were defined and implemented using ATL. In (Roser, 2006) the model transformations from business level ARIS models to platform-independent ICT system models have been described.

The potential of the intelligent support and recommendations for the enhancement of process modelling has been recently recognized (Smirnov, 2009). However, the few implemented modelling support tools have very restricted functionalities. Existing work in the area of intelligent support and recommendations for the process modelling can be split up in 2 categories: 1) methods for process analysis incl. model checking; 2) auto-completion of models (e.g. auto-completion of model elements identifiers, pattern recognition, auto-completion based on the methods of information retrieval).The most of implemented modelling support functionalities are used for conformity checking of models and based on auto completion of model elements identifiers (e.g. process editor Signavio (Fellmann, 2010)).

Another popular type of modelling support presents the model syntax checking. In the aforementioned Signavio the syntax checking has been implemented for EPC, BPMN 1.2 /2.0. With model checking different other model parameters like performance, reliability, security, maintenance, portability of models can be checked (Rech, 2009). The different information criteria can be also aggregated and used for the building of key figures like the similarity of models or the coverage rate of domain by one model (Fellmann, 2010).

More advanced modelling support systems contain functionality for validation and verification of process models as well as business process optimization. There are a lot of examples in the literature for the analysis of processes designed with BPMN by means of Petri net, which present the formal description of processes (e.g. (Desel, 2000)). So, (Raedts, 2007) has used BPMN-to-Petri net transformation for the validation and verification of industrial models. (Dijkman, 2007) has described an approach for the semantic correctness checking of process models with the help of Petri net.

Further type of auto completion functionality for modelling support systems is based on pattern recognition. Nowadays there are empirical researches dealing with the occurrence of patterns in the process models, e.g. in (Lau, 2009); (Smirnov, 2009); (Thom, 2009). There are also tools which are able to recognize patterns automatically and to propose the variants of completion like for example ProWAP (Thom, 2008).

The comprehensive concept of recommendation based editor for the business process modelling has been presented in (Hornung, 2008) and implemented by (Koschmider, 2010). In contrast to the described approaches this concept relates to the usual recommendation systems from information retrieval. This idea in the process modelling is that the designed model is compared with known models and the proposal for the extension or enhancement of this model is generated.

The described approaches have mostly indirect relation to service model transformation and recommendations for them, but all of them can be adapted to this field. Our proposal described in this work is to bring all together: model-driven approach, service orientation and intelligent (cross-level) modelling support concept.

3 MODEL TRANSFORMATIONS FOR SERVICE MODELS

3.1 CIM2PIM Transformation for SOA

Below we shortly sketch what kinds of models are defined on which abstraction level in the scope of this paper and how we intend to transform the models between those abstraction levels.

On the highest abstraction level CIM, business models encompass business rules, processes, services and other issues such as contracts involving humans and organizations to achieve business goals. These conform to the metamodel of CIMFlex prototype. The middle layer – PIM – contains the results of the proposal as transformation engines, extended SOA models, the standardized UPMS (SoaML).

The transformation engine should also support visualization of services in business models (provide transformation support both ways, top-down and bottom-up between CIM and PIM levels). This should provide a basis for service development for different application domains that covers the life-cycle of services from business goals and requirements to platform specific models for various platforms (Hahn, 2009).

Thus, the scope of this paper targets CIM modelling and bidirectional consistent CIM-to-PIM transformations.

3.2 Recommendations for Service Model Transformations

The background for the idea of intelligent modelling support system for transformations consists of 2 issues: 1) runtime of transformation for large models and 2) necessity for model verification and validation in time of their building. With the intelligent modelling support the user can get the information about how the changes made within one modelling level affect the process model within another modelling level, where the relation between modelling levels is transformation (see Figure 1). Change analysis can be used for round-trip engineering, as repeating of MDA cycles with model changes on different abstraction levels results in the entanglement of the models for the user (Delgado, 2010).

The modelling support can cover a big spectrum of aspects – from business level with process cost and durability analysis as well as the appropriateness of model changes up to technical level with the analysis of data consistency in the database or the changes of the methods in java class. A large analysis coverage can give the user much important information about models, but it is very risky to provide the good level of usability of such analysis tool at the same time (Fellmann, 2010). In our case study one can see that, using a relatively small set of analysis criteria, the analysis report can be very large.

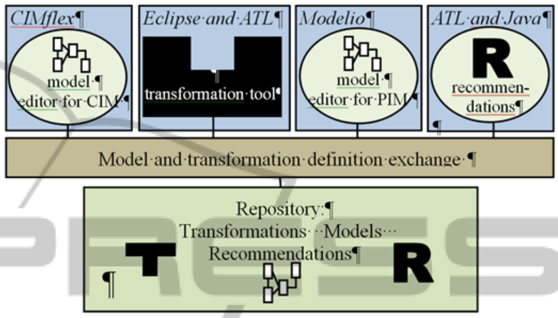

Figure 1: Implementation framework.

We examined all types of modelling support and considering also the specifics of technical implementation defined how different types of support can be implemented for the service model transformation in the MDA conceptually and on the technical level. The tool architecture is presented on the Figure 2. Transformation has been implemented in ATL, as a BPMN model editor CIMFlex and as a SoaML editor Modelio (www.modeliosoft.com) has been used correspondingly, recommendations have been implemented partly in ATL and in Java. We have chosen Eclipse and the compatible plug-ins for the implementation because the framework can be simply extended and provide the certain flexibility for developers (open source); moreover there are many components for Eclipse which can be used for the MDA.

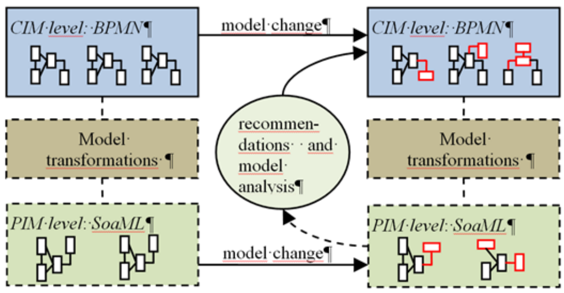

Figure 2: The idea of modelling cross-level support.

To restrict the recommendation coverage we have chosen for the implementation elements of 3 types of modelling support: 1) model checking (compliance with specification), 2) analysis of models by means of aggregated key figures and 3) analysis of model/process changes based on pattern recognition. The cross-level analysis can be done as follows: the basis input CIM is transformed in PIM and saved. Whenever the changes in CIM are done and they should be analysed on the PIM level, the new PIM is created, compared with the basis variant and the information about changes is return to the user.

The first idea here was to separate the comparison and analysis logic especially as there exist a lot of mature tools and approaches for the comparison of models. For our implementation framework the EMFT Compare looks like the most suitable tool for the model comparison. It provides generic support for any kind of metamodel in order to compare and merge models in the EMF framework. But because of many disadvantages like complex handling of profiles using EMFT Compare, no direct support of comparison of CIMFlex models, etc. we left this idea for the future work. The other existing tools for the model comparison operate on the low technical level (XML) and cannot be used for the implementation of our idea because we cannot provide 1:1 mapping of model elements repeating transformation between levels. The comparison of models within the MDA should be done on the level of model elements. To provide the high level of usability for our modelling support tool the most important details of models for the different types of analysis were chosen as the base for model analysis (usually it depends on the end user demand). For the model analysis by means of different key figures the model transformation has already been used. Many implemented examples of transformation of type model-to-measure can be found in the ATL transformations Zoo or for example in (Vépa, 2005). Such analysis is a time consuming solution due to the metamodel for the key figures and model-to-measure transformation.

All known approaches can be applied only for existent models and thus not very efficient for the in time modelling cross-level support. The approach which we have developed has a technical background and it is based on ATL Queries. Query is a special type of routines to compute a primitive value, such as a string or an integer, from source models. The advantage of the usage of queries for the cross-level model analysis is obvious: the query-logic is separated from transformation logic; the comparison of models is done on the model elements level; it is not complicated to describe the cross-level model changes only using the source models. The last statement is based on the property of queries to use the same elements of code as the transformation; therefore, this approach minimises the development effort and error rate of the analysis results. The remaining issue in this case is the interpretation of the analysis data and providing it the user in intelligent way.

4 MODEL TRANSFORMATIONS APPLICATION

4.1 Instantiation for BPMN2SoaML

To illustrate our concept we developed bidirectional BPMN-to-SoaML transformation in ATL and showed how the recommendations for the intelligent modelling support for transformation can be implemented using the selected tools.

The BPMN and SoaML models belong to the different abstraction levels, have different objects and key aspects. Furthermore, the complexity level of BPMN and SoaML is not identical. The study of (Recker, 2009) has shown that the complexity of full BPMN specification is higher as the complexity of full UML specification. Hence, different restrictions are inevitable for BPMN-to-SoaML transformations. The description of mapping rules for BPMN-elements into the corresponding UML/ SoaML elements presented in (Hahn, 2010) as well as in SHAPE project documentation. We enhanced and specified the transformation rules more precisely and adjusted them to the newer BPMN and SoaML/UML specifications.

To illustrate our transformation and following transformation support based on recommendations we chose the business case “Voyage travel agency” (presented at ECMFA’10 as a tutorial “Service Modeling with SoaML”: http://www.ecmfa-2010.org/index.php/tutorials),which includes 4 lanes representing the client, the partner, the payment centre and the travel agency itself. The client gets information about the current travel offer and chooses a travel. The client should provide personal information to the travel agency and confirm the order. Then the agency checks the client solvency with the collaboration of the payment centre and if the check is successful makes the reservations of travel for the client.

The specific feature of our transformation is that the CIM model built in BPMN contains 3 different views: the structural, behavioural and data view.

This separation means that our BPMN-to-SoaML transformation contains 3 different transformation types in one and the result will be consolidated in one resulting target model. The challenge of such transformation was how to integrate all views in one target model. The mapping rules from (Elvesæter, Panfilenko, 2010); (Hahn, 2009); (Hahn, 2010) include no integration aspects. Therefore, we have defined view integration rules, which are collected in the Table 1.

The target SoaML model contains also 3 views, but indirectly in contrast to the source BPMN model. These views can be seen in Modelio in 3 different diagrams: structural model as the SoaML participant diagram, behavioural model as the UML activity diagram and the data model as UML class diagram.

Table 1: The integration rules for SoaML models.

View	SoaML/UML element	superordinate element
Structural	Model	
	ServiceArchitecture	Model
	Participant	Model
	ServiceContract	Model
	ServicePoint	Participant
	RequestPoint	Participant
Behavioural	Activity	ServiceArchitecture
	Activity Partition	Activity
	other behavioural elements	Activity Partition
Data	All data elements	Participant

All of the services from the BPMN model appear in the SoaML model. The graphical correspondence of the models is not evident on the level of services because of incomplete compatibility of used tools, but the correspondence of structure of models on the internal XML level is satisfactory.

We don't cite the transformation of BPMN behavioural view into UML activity model in this paper, because this transformation is trivial and was described and implemented by the other authors (e.g. (Lemrabet, 2010)). The only difference between our and the other BPMN-to-SoaML transformations is the mapping set and rules.

The second part of our transformations is the inverse transformations. As mentioned above, a lot of researchers are in doubt about the possibility of PIM-to-CIM transformation. The challenge is that whenever the business models are transformed into service models, only the structural information is transformed, but not dynamical. The bidirectionality is only obtainable if the expression power of the both languages is identical (Kleppe, 2003).

In the case of bidirectional transformations the information is either lost (if the transformation function is a surjective function) or added (if the transformation function is an injective function) after the transformation step; to define a bijective model transformation function is almost impossible, particularly for the model transformations used in industry. There exists a method to solve this problem by using a tracing-technique, but in general it is hard solvable (Küster, 2004). In our research we have determined that the loss of information happens while changing the modelling level, but it can be reduced to the minimum.

In our scenario there could be the following information loss while transforming BPMN models into SoaML models (CIM-to-PIM):

- the direction of associations from DataObjects to Tasks in patterns Lane1→Lane2 und Role1→Role2(table 7 in (Hahn, 2010) und 9 in (Hahn, 2009));
- the relationship between Roles and Tasks from the pattern Role1→Role2;
- the type of Gateways;
- the source and target of Message Flows between Pools (table 27 in (Hahn, 2009)).

The loss of information is the cause of incomplete transformation definition, i.e. not every element from the CIM has a corresponding element on the PIM level. On the conceptual level we don't need to take into account all of the details defining the transformation rules, but in implementation stage these details can be important and must be included in the set of transformation rules. In our work we determined that using the appropriate traceability mechanism it is possible to achieve the complete restorable mapping.

We used the ATLAS transformation language, which does not support the extern traceability directly. One method of ensuring the traceability support in ATL was provided by (Jouault, 2005). The traceability information is considered as a special model and is described by means of traceability metamodel. This method was implemented as a special program – TracerAdder. It allows automating the extern traceability partially, because the program works in a special refining mode which is not fully supported by ATL yet. Nevertheless, this method has a big potential because the tracing and transformation logic are independent from each other as well as no additional language constructs or ATL engine modifications are needed, therefore its implementation is only the matter of time.

The similar technique can be implemented using profiles. This method was developed for UML-to-UML transformations (Vanhooff, 2005), but can be also adjusted for the other transformations.

In our concrete case we detected also another type of information loss which deals with the concept of "3 views"-representation of models described above. In BPMN models these views are integrated, whereas in UML/SoaML models they are completely independent. A lot of model elements can be assigned to different views, and therefore they are contained in the resulting SoaML models twice or even threefold (e.g. DataObject "ClientDefinition" appears in all three views), i.e. the target models of BPMN-to-SoaML transformation contain much more objects as the source models. This should be taken into account by the inverse SoaML-to-BPMN transformations. Because there is no relationship between such elements having "the same origin" in the SoaML, we need to think about the enhanced tracing technique which should be applied to all elements of models.

To sum up, the problem of the information loss by the bi-directional CIM-to-PIM transformations is solvable with one of the described traceability methods, but it still requires the adjustment of the whole concept and architecture.

4.2 SHAPE Project Case Study

Evaluation method in this paper refers to the evaluation conducted within the scope of the SHAPE project (Elvesæter, 2010). Evaluation should be split into two parts due to the nature of software development in companies like Saarstahl (www.saarstahl.com), having their own IT-department responsible for business process modelling. Firstly, a system architect asserts whether the solution proposed works and if it helps. Secondly, management has to assert the proposed solution fulfils the company guidelines for ROI. The evaluation conducted in SHAPE project is a subject to time and resource constraint and thus restricts the complete evaluation volume to performance measures definition, which in turn provides basis for answering the questions whether the solution works and whether it helps the development.

The full description of the Saarstahl use case realisation can be seen in (Elvesæter, 2010), whereas the scope of this paper is on the CIM level modelling and CIM-to-PIM transformations with recommendations for the latter. As for these the evaluation report states that on the CIM level CIMFlex allows modelling of business processes in an abstract manner. As for CIM-to-PIM transformations, these generate a skeleton SoaML model that needs further manual refinements. The recommendations for the model transformations are the research point, especially given attention in this paper. Using these techniques Saarstahl was able to model its business processes on different abstraction levels, whereas the produced source code and system reflect the business models on the top abstraction level, which in turn proves that the proposed technology works. Another part of the use case was to wrap the existing legacy systems behind the web services, which also has been successfully evaluated. The modelling as described and the wrapping of the legacy systems lead to increased interoperability of the complete IT landscape at Saarstahl, which proves the consistency of the SHAPE technology. The complete list of performance measures that provide basis for answering the question about ROI can be seen in (Elvesæter, 2010).

5 DISCUSSION AND IMPLICATIONS

Of course, the described work is a trial version of BPMN-to-SoaML transformation to evaluate our concept of intelligent cross-level modelling support and review the gaps in it. The described transformations are implemented with the pre-defined tools and big restrictions to be solved in the future:

1) The set of mapping rules does not cover all elements of metamodels which are the basis of the used modelling editors. The consequence is that the source models should be manually checked and restricted to include only the elements described in transformation rules. It causes the additional difficulties in testing and validation of transformations particularly for the models with a lot of elements.

2) The mapping rules are usually defined on the conceptual level and do not contain a specific technical information. This requires extending the set of transformation rules with additional ones, which complicates the understanding of the transformations. We defined additional rules (see Table 1) to achieve the unique structural representation of PIMs in our transformation, because the initial set of mapping rules did not includ the integration aspect of different view in SoaML models.

3) The transformation rules of type pattern-to-element require a lot of implementation effort. We used the linear search approach to identify the patterns and tested our transformation on the models containing not more as 120 elements. Probably it can be necessary in the future to think of a special optimisation algorithm for larger models.

4) Further problem with patterns consists in the interpretation of patterns combination (Elvesæter, Panfilenko, 2010), (Hahn, 2009), (Hahn, 2010). We detected only some combinations which require more precise description, but it can be identified more. So it is necessary to include the interpretation of patterns combinations in the mapping rules or define the patterns more exactly.

5) The implemented model transformation and generation of the recommendations for the transformations are only partly automated due to the existing tools not achieving such level of maturity yet. In our case the testing tool, the validation and verification environments are missing. It is possible, though, to find more appropriate technologies to automate the desired concept.

6) The implemented recommendations cover only some aspects of possible intelligent cross-level modelling support. It is possible to extend the number of recommendations by including other types of model analysis or extending the existent recommendations for the bigger set of model elements.

7) The transformations and recommendations were implemented with big restrictions, which are the cause of compatibility problems between different tools we used in our research. These problems are solvable by means of different techniques indeed, but make the transformations very complicated to understand, to modify and to reuse.

6 CONCLUSIONS AND FUTURE WORK

MDA is a powerful concept which can be used in many fields like process analysis, process optimisation, model validation, model driven testing, etc. However, there are a lot of problems (e.g. model transformation) on the way of the MDA implementation mentioned in the literature and verified in our work as well.

The technology for the implementation of model transformations should be chosen very carefully. All known transformation languages have such level of maturity at the moment that they cannot satisfy all requirements for the model transformations in the MDA. Further aspect, which makes the development of model transformations time-consuming, is that there is no integrated implementation framework for the transformation, i.e. validation, testing and debugging should be done manually. Moreover, all of the existent model editors do not support the automatic arrangement and alignment of the graphical elements for the exported models.

Our concept, including the implementation technique, allows for strict distinction between the transformation and analysis logics, but they use the same functions at the same time, which guarantees the synchronisation of model transformations and the analysis of transformed models. This way does not require any complicated modifications in other tools. However, the recommendations coverage should be still defined. Once all necessary formalisms are defined, the model analysis can be easily applied on different modelling levels as well as in the cross-level field.

The unique architecture for the transformation and cross-level recommendations in the MDA should still be elaborated. The concept of model validation should be implemented; a big advantage would be an appropriate traceability mechanism for the transformation as well as an automated test environment. The smooth interaction of all necessary components still requires a lot of work on the both conceptual and technical levels.

ACKNOWLEDGEMENTS

This research was co-funded by the European Union in the frame of the SHAPE FP7 project (ICT- 2007-216408). The authors would like to express their acknowledgments to SHAPE colleagues.

REFERENCES

ATLAS Group (INRIA and LINA) 2005: “ATL Transformation Examples. The KM3 to Metric ATL transformation”, Version 0.1.

De Castro V., Marcos E., Vara J. M., 2011: “Applying CIM-to-PIM model transformations for the service-oriented development of information systems,” in *Information and Software Technology, Volume 53, Issue 1*, pp 87-105.

Delgado A., García-Rodríguez de Guzmán I., Ruiz F., Piattini M. 2010: “Tool support for Service Oriented development from Business Processes,” in *MOSE2010* in conjunction with the *TOOLS2010*, Málaga, Spain.

Desel, J.; Erwin, T. 2000: “Modeling, Simulation and Analysis of Business Processes,” in: van der Aalst, W. et al. (eds.): *Business Process Management, LNCS 1806*. Berlin Heidelberg: Springer, pp. 129-141.

Dijkman, R.M.; Dumas, M.; Ouyang, C. 2007: “Formal Semantics and Analysis of BPMN Process Models using Petri Nets,” Technical Report 7115, Queensland University of Technology, Brisbane. http://eprints.qut.edu.au/archive/00007115/.

Elvesæter, B.; Landre, E.; Hahn, C.; Jacobi, S. 2010: Case Study Execution and Validation”. SHAPE-Project, Deliverable D1.4, version 1.0/

Elvesæter, B.; Panfilenko, D.; Jacobi, S.; Hahn, C. 2010: “Aligning Business and IT Models in Service-Oriented Architectures using BPMN and SoaML,” in: *MDI10*, Bézivin, J.; Soley R.M.; Vallecillo A.(eds.), ACM/IEEE *MoDELS10*, Oslo, Norway. New York: ACM Press, pp. 61-68.

Thomas Erl. 2007. Soa: Principles of Service Design (First ed.). Prentice Hall Press, NJ, USA.

Fellmann, M.; Zarvic, N.; Thomas, O. 2010: „Unterstützung von Modellierungsleistungen durch semantische Technologien,“ in: *Dienstleistungsmodellierung 2010. Interdisziplinäre Konzepte und Anwendungsszenarien*, Thomas, O.; Nüttgens, M. (eds.). Heidelberg: Springer, pp. 70-80.

Hahn, C.; Panfilenko, D. 2009: “Model transformations from business models to UPMSHA – Initial version”. SHAPE-Projekt, Deliverable D5.2, Version 0.5. 01.12.2009. p. 7-10.

Hahn, C.; Panfilenko, D.; Fischer, K. 2010: A Model-Driven Approach to Close the Gap between Business Requirements and Agent-Based Execution.

Hornung, T., Koschmider, A., Lausen, G. 2008: “Recommendation based process modelling support: Method and user experience,” in: *ER 2008. LNCS 5231*, Li, Q., Spaccapietra, S., Yu, E., Olivé, A. (eds.). Heidelberg: Springer.

Jeary, S.; Fouad, A.; Phalp, K. 2008. “Extending the Model Driven Architecture with a preCIM level,” in TOOLS EUROPE Workshop, 1st International Workshop on Business Support and MDA (MDABIZ), Zurich, p.2.

Jouault, F. 2005: Loosely Coupled Traceability for ATL. http://www.sciences.univ-nantes.fr/lina/atl/.

Kahl, T.; Kupsch, F. 2005. „Transformation und Mapping von Prozessmodellen in verteilten Umgebungen mit der ereignisgesteuerten Prozesskette,“ in: *EPK 2005 - Geschäftsprozessmanagement mit Ereignisgesteuerten Prozessketten - 4. Workshop der Gesellschaft für Informatik e.V. (GI) - und Treffen ihres Arbeitkreises „Geschäftsprozessmanagement mit Ereignisgesteuerten Prozessketten“*, Nüttgens, M.; Rump, F. J. (eds.), Hamburg, p. 3.

Kleppe, A.; Warmer, J.; Bast, W. 2003.„MDA Explained, The Model Driven Architecture: Practice and Promise”, Addison-Wesley.

Koch, N. 2006. “Transformation Techniques in the Model-Driven Development Process of UWE,” in: ICWE'06 Workshops. Palo Alto, CA.

Koschmider, A.; Oberweis, A. 2010: “Designing Business Processes with a Recommendation-Based Editor,” in: *Handbook on Business Process Management, Band 1. Introduction, Methods, and Information Systems*, vomBrocke, J.; Rosemann, M. (eds.). Berlin Heidelberg: Springer, pp. 299-313.

Küster, J. M.; Sendall, S.; Wahler, M. 2004: “Comparing two model transformation Approaches,” in: *UML 2004, Workshop OCL and Model Driven Engineering*, Proceedings, Lisbon, Portugal.

Lau, J. M.; Iochpe, C.; Thom, L. et al. 2009: “Discovery and Analysis of Activity Pattern Cooccurrences in Business Process Models,” in: *Proc. 11th International Conference on Enterprise Information Systems (ICEIS'09), Volume on Information Systems Analysis and Specification*, Milan, Italy, LNBIP 24, Filipe, J; Cordeiro, J. (eds.). Berlin Heidelberg: Springer, pp. 83-88.

Lemrabet, Y.;Touzi, J.; Clin, D.; Bigand, M.; Bourey, J.-P., 2010. “Mapping of BPMN models into UML models using SoaML profile,” in: *MOSIM’10*, Hammamet, Tunisia.

Petrasch, R.; Meimberg, O. 2006. „Model DrivenArchitecture. Eine praxisorientierte Einführung in die MDA,“ Heidelberg: dpunkt.verlag.

Raedts, I.; Petkovic, M.; Usenko, Y. et al. 2007: “Transformation of BPMN models for behaviour analysis,” in: *Proceedings of MSVVEIS*, Augusto, J.; Barjis, J.; Ultes-Nitsche, U. (eds.). INSTICC Press, pp. 126-137.

Rech, J.; Schmitt, M. 2009: “Embedding Information about Defects, Decisions, Context, Quality, and Traceability in CIM- and PIM-level Software Models,” in: *Information and Software Technology* (IST), p. 2.

Recker, J. C.; zurMuehlen, M.; Siau, K. et al. 2009: “Measuring method complexity: UML versus BPMN,” in: *Proc. AMCIS09*, San Francisco, California. p. 7.

Rodríguez, A.; Fernández-Medina, E.; Piattini, M. 2007: “Towards CIM to PIM Transformation: From Secure Business Processes Defined in BPMN to Use-Cases,” in: *5th Int. Conf. on BPM, LNCS 4714*, pp. 408-415.

Rodríguez, A.; Fernández-Medina, E.; Piattini, M. 2008: "CIM to PIM Transformation: A Reality,” in *Research and Practical Issues of Enterprise Information Systems II IFIP International Federation for Information Processing*, Volume 255, pp. 1239-1249.

Roser,S., Bauer,B., Muller,J. 2006: “Model- and Architecture-Driven Development in the Context of Cross-Enterprise Business Process Engineering,” in: *SCC’06*.

Smirnov, S.; Weidlich, M.; Mendling, J. 2009: “Action Patterns in Business Process Models,” in: *Service-Oriented Computing. LNCS 5900*, Baresi, L.; Chi, C.-H.; Suzuki, J. (eds.). Heidelberg: Springer , pp. 115-129.

Thom, L.; Reichert, M.; Chiao, C. M. et al. 2008: “Applying Activity Patterns for Developing an Intelligent Process Modeling Tool,” in: *ICEIS'08*, Barcelona, Spain.

Thom, L. H.; Iochpe, C.; Reichert, M. et al. 2009: "Activity patterns in process-aware information systems: Basic concepts and empirical evidence," in: *IJPIM*, Vol. 4, No. 2, pp. 93-110.

Touzi J., Benaben F., Pingaud H., Lorré J.P. 2009: "A model-driven approach for collaborative service-oriented architecture design," in: *International Journal of Production Economics*, Volume 121, Issue 1, pp. 5-20.

Vépa, E. 2005: "ATL Transformation Examples. UML2 to Measure," Version 0.1. http://www.eclipse.org/m2m/atl/atlTransformations/.

Vanhooff, B.; Berbers, Y. 2005: "Supporting Modular Transformation Units with Precise Transformation Traceability Metadata," in: *ECMDA-TW 2005, Proceedings*. Nuremberg, p. 20.