Enterprise Model Based UML Interaction Overview Model Generation Process

Audrius Lopata, Ilona Veitaite(✉), and Neringa Zemaityte

Department of Informatics, Kaunas Faculty of Humanities,
Vilnius University, Muitines g. 8, 44280 Kaunas, Lithuania
{Audrius.Lopata, Ilona.Veitaite,
Neringa.Zemaityte}@khf.vu.lt

Abstract. The main scope of the research is to analyse Unified Modelling Language (UML) models generation process from Enterprise Model (EM) in Information Systems (IS) development process by using knowledge-based subsystem. The knowledge-based subsystem is proposed as an additional computer aided software engineering (CASE) tool component to avoid IS development process based on empirics. For comprehensible perception there is also presented relation between EM and ontologies and its use in generation process.

As the result of this part of research transformation algorithms are presented and described. These algorithms are capable of whole UML models elements generation from Enterprise Model. Example of UML Interaction Overview model generation illustrates full process.

Keywords: Enterprise Modelling · Ontology · Knowledge-based · CASE · IS engineering · UML · Interaction Overview model

1 Introduction

In a modern world software development and software applications are becoming more complex and demanding. Developers, analysts, engineers, researchers are creating and seeking for new techniques and procedures to streamline software engineering processes to ensure shorter development time and reduce costs by re-using different components. The development of software systems is a complex activity which may imply the participation of people and machines (distributed or not). Therefore, different stakeholders, heterogeneity and new software features make software development a heavily knowledge-based process [1, 11].

In modern day enterprise engineering, it is paramount that Enterprise Models are grounded in a well-defined, agreed-upon Enterprise Architecture that captures the essentials of the business, IT, and its evolution. Enterprise architectures typically contain different views (e.g. Business, Information, Process, Application, Technical) on the enterprise that are developed by distinct stakeholders with a different background and knowledge of the business. Consequently, the developed Enterprise Models that populate these views are hard to integrate. A possible solution for this integration problem is using a shared terminology during the development of these different views [2].

W. Abramowicz et al. (Eds.): BIS 2016 Workshops, LNBIP 263, pp. 69–78, 2017.
DOI: 10.1007/978-3-319-52464-1_7

Such explicit formal representations, often materialized in the form of ontology – in a business context called an enterprise-specific ontology – provide a myriad of advantages. Ontologies are shared views of domains. They provide conceptualizations that are agreed upon by participants in collaborative action and decision making. The explicit existence of such shared perspectives makes it possible for both people and programs to collaborate by ensuring that everybody makes the same distinctions and uses the same terms with the same meaning [18]. On an intra-organizational level, they ensure model re-usability, compatibility and interoperability, and form an excellent basis for enterprise-supporting IT tools, such as Enterprise Resource Planning (ERP) systems, business intelligence (BI) tools or information systems (IS), for which they serve as common terminology. On an inter-organizational level, they facilitate interoperability, cooperation and integration by allowing formal mappings between, and alignment of separately developed Enterprise Models [12].

2 Enterprise Modelling and Ontologies Relation

An Enterprise Model is a computational representation of the structure, activities, processes, information, resources, people, behaviour, goals and constraints of a business, government, or other enterprise. It can be both descriptive and definitional - spanning what is and what should be. The role of an Enterprise Model is to achieve model-driven enterprise design, analysis and operation [6, 18]. Enterprise Modelling is an activity where an integrated and commonly shared model of an enterprise is created [7, 12, 28]. The resulting Enterprise Model comprises several sub-models, each representing one specific aspect of the enterprise, and each modelled using an appropriate modelling language for the task at hand. For example, the Enterprise Model may contain processes modelled in BPMN, data modelled in ER and goals modelled in n*. The Enterprise Model is thus developed by several enterprise engineers, and aggregates all information about the enterprise. As a result, Enterprise Models without homogenized underlying vocabulary suffer interoperability and integration problems [12, 25]. An Enterprise Model can be developed for single or more different purposes. Few Enterprise Modelling formal purposes are presented [3, 20, 22]:

1. To capitalize enterprise knowledge and know how.
2. To illustrate relations and dependencies within the enterprise and with other enterprises, to achieve better control and management over all aspects.
3. To provide support to business process re-engineering.
4. To get a common and complete understanding of the enterprise.
5. To improve information management across organizational and application system boundaries and provide a common means for communication throughout the organization. Rationalize and secure information flows.
6. To provide operative support for daily work at all levels in the enterprise from top to bottom.
7. To control, co-ordinate and monitor some parts of the enterprise.
8. To provide support for decision making.
9. To provide support the design of new parts of the enterprise.
10. To simulate processes.

Ontology is a discipline rooted in philosophy and formal logic, introduced by the Artificial Intelligence community in the 1980s to describe real world concepts that are independent of specific applications. Over the past two decades, knowledge representation methodologies and technologies have subsequently been used in other branches of computing where there is a need to represent and share contextual knowledge independently of applications [22, 23].

Ontology engineering is a filiation of knowledge engineering that studies the methods and methodologies for building ontologies. In the domain of enterprise architecture, ontology is an outline or a schema used to structure objects, their attributes and relationships in a consistent manner. As in Enterprise Modelling, ontology can be composed of other ontologies. The purpose of ontologies in Enterprise Modelling is to formalize and establish the shared understanding, reuse, assimilation and dissemination of information across all organizations and departments within an enterprise. Also, an ontology enables integration of the various functions and processes which take place in an enterprise [10].

Using ontologies in Enterprise Modelling offers several advantages. Ontologies ensure clarity, consistency, and structure to a model. They promote efficient model definition and analysis. Generic enterprise ontologies allow for reusability and automation of components. A common ontology allows to ensure shared understanding, clearer communication, and more effective coordination among the various divisions of an enterprise. These lead to more efficient production and flexibility within the enterprise [24].

3 Transformation Algorithm

The computerized IS engineering specific methods are developed based on common requirements, which systematize the selected methodology. Computerized knowledge-based IS engineering project management basis is CASE system knowledge-based subsystem. CASE system's knowledge-based subsystem's core component is knowledge base, which essential elements are enterprise meta–model specification and Enterprise Model for certain problem domain [4, 8, 24]. Knowledge-based subsystem is one more active participant of IS engineering process beside analyst, whose purpose is to verify results of IS life cycle phases [5, 9].

Knowledge-based CASE systems holding substantial components, which organize knowledge: knowledge-based subsystem's knowledge base, which essential elements are enterprise meta–model specification and Enterprise Model for certain problem domain [7, 13, 16]. Figure 1 presents knowledge-based subsystem connection to the Enterprise Model and Enterprise Meta–Model inside CASE tool presented as Sequence diagram.

Information system design methods indicate the continuance of systems engineering actions, i.e. how, in what order and what UML models to use in the design process and how to fulfil the process. Association between UML models and EM is realized through the transformation algorithms and presented in Fig. 2 [14, 15].

Enterprise Model as organization's knowledge repository enables generate UML models with the help of transformation algorithms. Enterprise meta-model holds

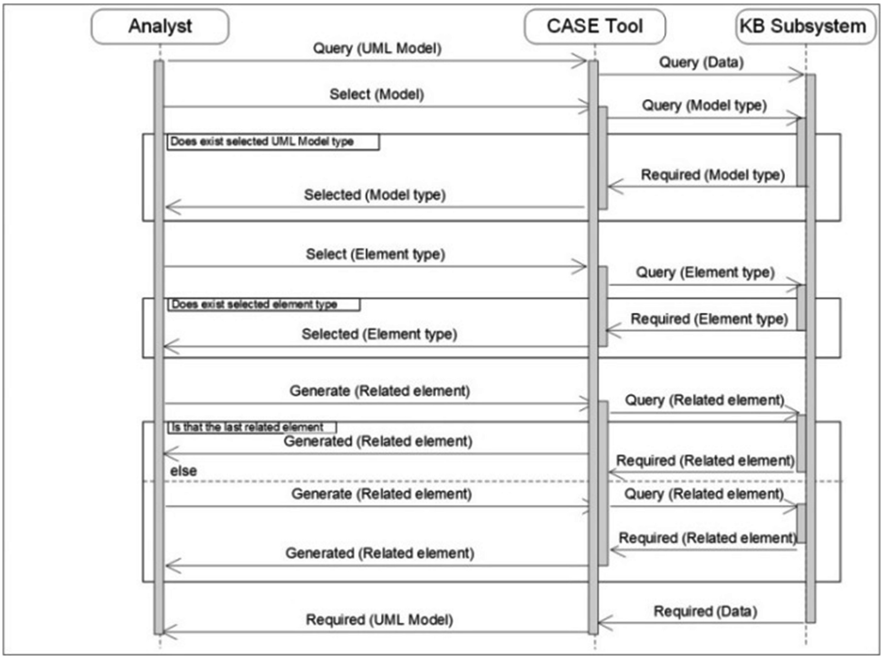

Fig. 1. Knowledge-based subsystem connection to the Enterprise Model and Enterprise Meta–Model inside CASE tool presented as Sequence diagram

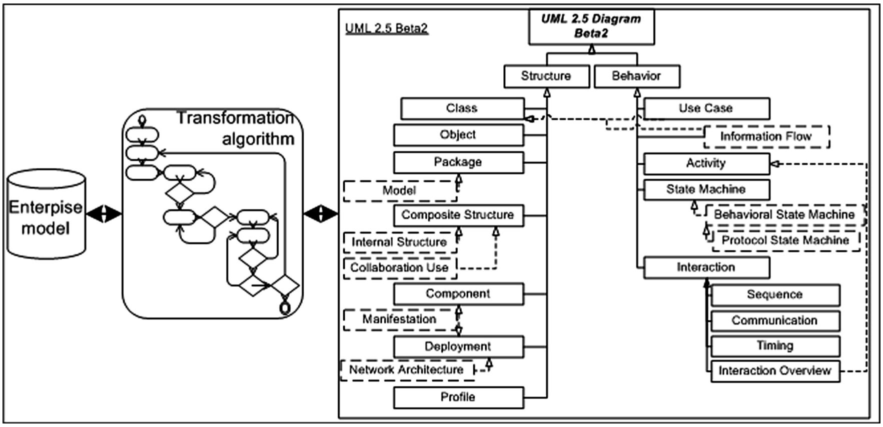

Fig. 2. UML models generation by using the transformation algorithm [21, 27]

essential elements of business modelling methodologies and techniques, which ensures a proper UML models generation process [17, 19].

Presently, used CASE system's Enterprise Models constitution is not verified by formalized criteria. Enterprise Models have been formed in compliance with the notations. However, their composition has not been proved by the characteristics of the specific domain area [26, 27].

In IS engineering all design models are fulfilled on the basis of the empirical expert experience. Experts, who participate in the IS development process, do not gain enough knowledge, and process implementation in requirements analysis and specification phases can take a too long time. Enterprise meta–model contains essential elements of business modelling methodologies and techniques, which insures a suitable UML diagrams generation process [26, 27].

Figure 3 presents top level transformation algorithm for Enterprise meta–model based UML models generating process. Main steps for generating process are identifying and selecting UML model for generating process, identifying starting elements for the selected UML model and selecting all related elements, reflecting Enterprise Model elements to UML model elements and generating the selected UML model [21, 26, 27].

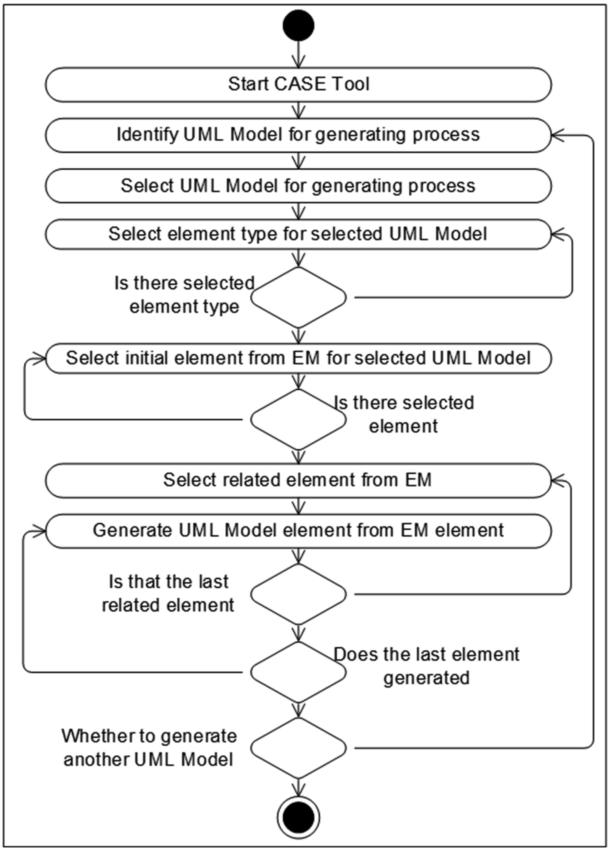
Start CASE Tool
Identify UML Model for generating process
Select UML Model for generating process
Select element type for selected UML Model
Is there selected element type
Select initial element from EM for selected UML Model
Is there selected element
Select related element from EM
Generate UML Model element from EM element
Is that the last related element
Does the last element generated
Whether to generate another UML Model

Fig. 3. Transformation algorithm of EM based UML model generation process

4 UML Interaction Overview Model Transformation

UML Interaction Overview diagram determines interactions through a variant of activity diagrams in a manner that maintains overview of the control flow. Interaction Overview model concentrate on the overview of the flow of control where the nodes are interactions or interaction uses. The lifelines and the messages do not perform at this overview level. UML Interaction Overview model combines elements from activity and interaction diagrams [21]:

- the following elements of the activity diagrams could be used on the Interaction Overview diagrams: initial node, flow final node, activity final node, decision node, merge node, fork node, join node;
- the following elements of the interaction diagrams could be used on the Interaction Overview diagrams: interaction, interaction use, duration constraint, time constraint.

Main steps of UML Interaction Overview model generation from Enterprise Model transformation algorithm are (see Fig. 4): selecting Interaction Overview model for generating process, identifying initial element, selecting element's type for chosen model, selecting related model elements and generating model.

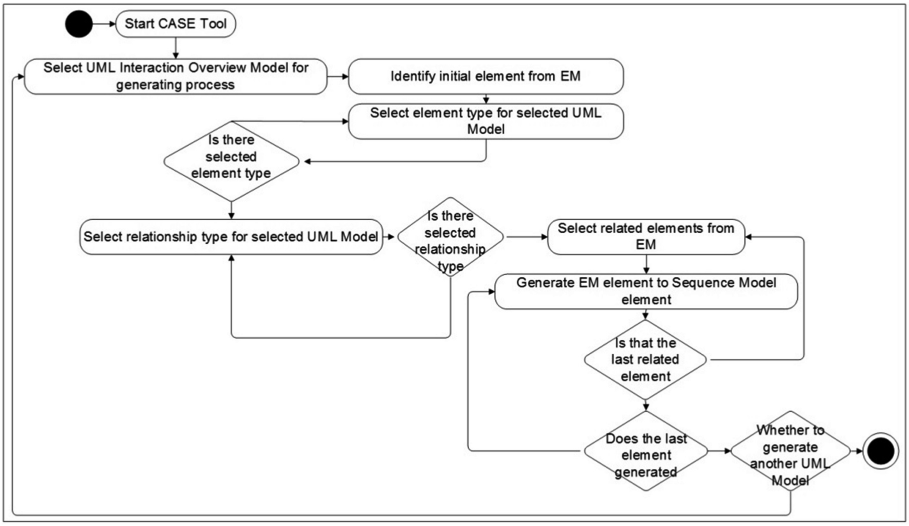
Start CASE Tool
Select UML Interaction Overview Model for generating process
Identify initial element from EM
Select element type for selected UML Model
Is there selected element type
Select relationship type for selected UML Model
Is there selected relationship type
Select related elements from EM
Generate EM element to Sequence Model element
Is that the last related element
Does the last element generated
Whether to generate another UML Model

Fig. 4. Transformation algorithm of EM based UML Interaction Overview model generation process

Table 1 presents UML Interaction Overview model elements generated from Enterprise Model. Frame as Interaction model element is generated from EM Actor element, Interaction Use as Interaction model element is generated from EM Information Activity, Initial Node, Decision Node, Merge Node, Final Node as Activity model elements are generated from EM Business Rules elements and Decision Guard as Activity model element is generated from EM Information Flow element.

Table 1. EM and Online service ordering UML Interaction Overview model elements

EM		UML Interaction Overview model						
		Frame (Interaction model element)	Interaction use (Interaction model element)	Initial node (Activity model element)	Decision node (Activity model element)	Merge node (Activity model element)	Final node (Activity model element)	Decision guard (Activity model element)
Actor		+						
Event								
Process	Material input							
	Material output							
Function	Business rules			+	+	+	+	
	Information flow							+
	Information activity		+					

Figure 5 presents an example of UML Interaction Overview model. The necessary elements through transformation algorithms were received from CASE tool's knowledge-based subsystem's Enterprise Model, where all knowledge of subject area is stored. In this figure it is clearly seen all necessary UML Interaction Overview model elements generated from Enterprise Model.

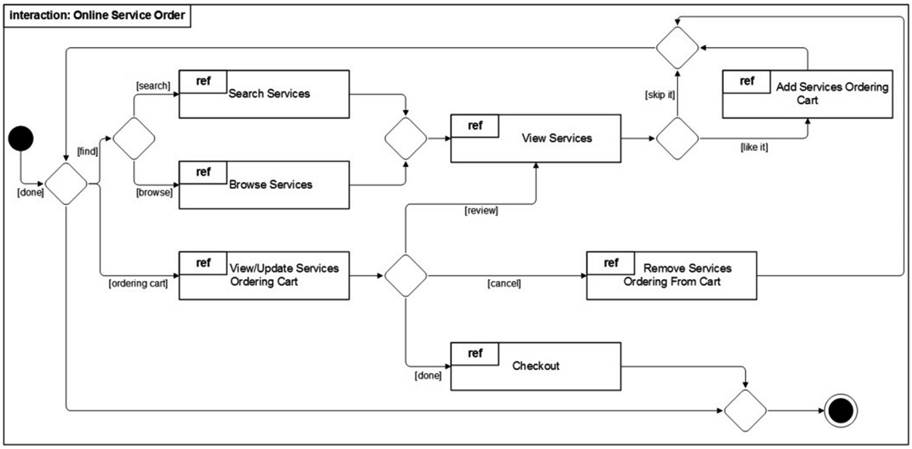

Fig. 5. UML Interaction Overview model example: online service order

5 Conclusions

Computer aided IS engineering is based on empiric and IS development life cycle stages are fulfilled on the basis of the expert's experience. A large part of the CASE tools design models are generated only partially, and complete realization is possible only non-automatic and with experts participation. Today IS engineering should be based on knowledge. In this way, knowledge-based IS engineering computerized IS development activities are executed using the subject area knowledge, which is stored in the knowledge base of CASE tool repository.

In order to decrease the influence of empirical factors on IS development process, the decision was made to use knowledge-based IS engineering approach. The main advantage of this approach is the possibility to validate specified data stored in EM against formal criteria, in that way decreasing the possible issues and ensuring more effective IS development process compared to classical IS development methods.

Using ontologies in Enterprise Modelling offers several advantages. Ontologies ensure clarity, consistency, and structure to a model. They promote efficient model definition and analysis. Generic enterprise ontologies allow for reusability of and automation of components. Because ontologies are schemes or outlines, the use of ontologies does not insure proper Enterprise Model definition and analysis. Ontologies are limited by how they are defined and fulfilled. Ontology not always includes ability to cover all of the aspects of what is being modelled.

The paper deals with the generation process of UML models from EM options. Every element of UML model can be generated from the EM using CASE Tool knowledge base's subsystem and transformation algorithms. Method of UML model generation process from EM could implement full knowledge-based IS development cycle design stage. This is partially established by the example of online service ordering presented as UML Interaction Overview model elements generation.

References

1. Alonso, J.B.: Ontology-based software engineering, engineering support for autonomous systems. ASLab-ICEA-R-2006-016 v 0.1 Draft of 2006-11-15 (2006)
2. Bera, P., Burton-Jones, A., Wand, Y.: Guidelines for designing visual ontologies to support knowledge identification. MIS Q. **35**, 883–908 (2011)
3. Brathaug, T.A., Evjen, T.Å.: Enterprise Modeling. SINTEF, Trondheim (1996)
4. Butleris, R., Lopata, A., Ambraziunas, M., Veitaite, I., Masteika, S.: SysML and UML models usage in knowledge based MDA process. Elektronika ir elektrotechnika **21**(2), 50–57 (2015). Print ISSN: 1392-1215, Online ISSN: 2029-5731
5. IEEE Computer Society: Guide to the Software Engineering Body of Knowledge SWEBOK. Version 3.0. Paperback ISBN-13: 978-0-7695-5166-1 (2014)
6. Gudas, S.: enterprise knowledge modelling: domains and aspects. Technological and economic development of economy. Baltic J. Sustain. **15**, 281–293 (2009)

7. Gudas, S.: Architecture of knowledge-based enterprise management systems: a control view. In: Proceedings of the 13th World Multiconference on Systemics, Cybernetics and Informatics (WMSCI 2009), 10–13 July 2009, Orlando, Florida, USA, vol. III, pp. 161–266 (2009). ISBN-10: 1-9934272-61-2 (vol. III). ISBN-13: 978-1-9934272-61-9 (vol. III)
8. Gudas, S.: Informacijos sistemų inzinerijos teorijos pagrindai. Vilniaus universiteto leidykla (2012). ISBN 978-609-459-075-7
9. Gudas, S., Lopata, A.: Meta-model based development of use case model for business function. Information Technology and Control, vol. 36, no. 3 (2007). ISSN 1392 – 124X 2007
10. Fadel, G., Fox, M., Gruninger, M.: A generic enterprise resource ontology. In: Proceedings of the 3rd Workshop on Enabling Technologies: Infrastructure for Collaborative Enterprises, pp. 117–128 (1994)
11. Force, S.E.T.: Ontology driven architectures and potential uses of the semantic web in systems and software engineering (2001)
12. Gailly, F., Casteleyn, S., Alkhaldi, N.: On the symbiosis between enterprise modelling and ontology engineering. In: Ng, W., Storey, V.C., Trujillo, J.C. (eds.) ER 2013. LNCS, vol. 8217, pp. 487–494. Springer, Heidelberg (2013). doi:10.1007/978-3-642-41924-9_42
13. Lopata, A.: Veiklos modeliu grindziamas kompiuterizuotas funkcinių vartotojo reikalavimų specifikavimo metodas, Disertacija (2004)
14. Lopata, A., Veitaite, I., Gudas, S., Butleris, R.: CASE tool component – knowledge-based subsystem. UML diagrams generation process. Transformations Bus. Econ. **13**, no. 2B(32B), pp. 676–696 (2014). ISSN: 1648 - 4460
15. Lopata, A., Veitaite, I.: UML diagrams generation process by using knowledge-based subsystem. In: Abramowicz, W. (ed.) BIS 2013. LNBIP, vol. 160, pp. 53–60. Springer, Heidelberg (2013). doi:10.1007/978-3-642-41687-3_7
16. Lopata, A., Ambraziunas, M., Gudas, S., Butleris, R.: The main principles of knowledge-based information systems engineering. Electron. Electr. Eng. **11**, no. 1(25), 99–102 (2012). ISSN 2029-5731
17. Lopata, A., Ambraziunas, M., Gudas, S.: Knowledge based MDA requirements specification and validation technique. Transformations Bus. Econ. **11**, no. 1(25), pp. 248–261 (2012). ISSN 1648-4460
18. Fox, M.S., Barbuceanu, M., Gruninger, M., Lin, J.: An Organization Ontology for Enterprise Modelling. MIT Press, Cambridge, pp. 131–152 (1998). ISBN:0-262-66108-X
19. Morkevicius, A., Gudas, S.: Enterprise knowledge based software requirements elicitation. Inf. Technol. Control **40**(3), 181–190 (2011). ISSN 1392 – 124X
20. Khan, N.A.: Transformation of enterprise model to enterprise ontology. Master Thesis, Jonkoping, Sweden (2011)
21. OMG UML: Unified Modelling Language version 2.5. Unified Modelling (2012). http://www.omg.org/spec/UML/2.5/Beta2/
22. OMG ODM: OMG Formal Versions of Ontology Definition Metamodel (2014). http://www.omg.org/spec/ODM/1.1
23. Ostie, J.K.: An introduction to enterprise modeling and simulation (1996)
24. Perjons, E.: Model-driven process design. Aligning value networks, enterprise goals, services and IT systems. Department of Computer and Systems Sciences, Stockholm University, Sweden by US-AB, Stockholm (2011). ISBN 978-91-7447-249-3
25. Stirna, J., Persson, A., Sandkuhl, K.: Participative enterprise modeling: experiences and recommendations. In: Krogstie, J., Opdahl, A., Sindre, G. (eds.) CAiSE 2007. LNCS, vol. 4495, pp. 546–560. Springer, Heidelberg (2007). doi:10.1007/978-3-540-72988-4_38

26. Veitaite, I., Lopata, A.: Additional knowledge based MOF architecture layer for UML models generation process. In: Abramowicz, W. (ed.) BIS 2015. LNBIP, vol. 228, pp. 56–63. Springer, Heidelberg (2015). doi:10.1007/978-3-319-26762-3_6
27. Veitaitė, I., Ambraziūnas, M., Lopata, A.: Enterprise model and ISO standards based information system's development process. In: Abramowicz, W., Kokkinaki, A. (eds.) BIS 2014. LNBIP, vol. 183, pp. 73–79. Springer, Heidelberg (2014). doi:10.1007/978-3-319-11460-6_7
28. Vernadat, F.: UEML: towards a unified Enterprise Modelling language. Int. J. Prod. Res. **40**, 4309–4321 (2002)