Requirements Engineering, Supported by Ontology and Enterprise Modelling

Audrius Lopata, Neringa Makrickienė
Vilnius University, Kaunas Faculty, Institute of Applied Informatics
Muitinės g. 8, LT-44280, Kaunas, Lithuania
audrius.lopata@knf.vu.lt, neringa.makrickiene@knf.vu.lt

Abstract – nowadays, as information technology becomes more and more demanding, system become bigger in the terms of scope, the need in well established requirements becomes crucial. The analysis of an information system requirements should result in the establishment of well-defined functionalities and attributes agreed by the stakeholders. If the functionalities are defined as incomplete or incorrect, the software may not meet the expectations of users. In this article the importance of well prepared requirements is stated by analyzing and merging such technologies as Enterprise Modelling and Ontologies in the context of the MOF architecture. The basic concept of the upgraded MOF architecture is presented, as well as, the Enterprise Metamodel and Requirements Ontology are brought together to solve the problems in requirements engineering area.

Keywords – Enterprise Modelling, EM, MDD, MOF, Ontology, Requirements Engineering.

I. 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. Yet, it is well known that stakeholders requirements are standing at the root of software development process and it is critical component. IT professionals have already recognized the importance of correct requirements for successful results, because faults in requirements phase influence all phases of software development.

Requirements Engineering process recently evolved, because attentively developed requirements became crucial for successful projects. This process became very collaborative, including stakeholders from various areas with the aim to develop business domain into features and attributes of the software. Yet, stakeholders from different areas of knowledge, their communication skills and new software features make information systems development a heavily knowledge-based process [1, 8].

In a modern day enterprise engineering, it is important that Enterprise Models are developed in a well-defined Enterprise Architecture that captures the essentials of the business, IT, and its ever changing processes. Enterprise architectures typically contain different point of views (e.g. Information, Business area, Process, Application, Technical details and etc.) on the enterprise that is developed by different stakeholders with a variety background and knowledge about the business. Many authors think that, consequences of the developed Enterprise Models that populate these views are hard to integrate. Yet, one of the options, considered as a possible solution for this integration problem could be using a shared terminology during the development of such cases [2]. Shared terminology, often materialized in the form of ontology – in a business context called an enterprise-specific ontology – provide many advantages. Some authors describe ontologies as shared views of domains. Ontologies provide conceptualizations that are agreed upon by participants in collaborative action and decision making. As it is mentioned in [16], 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. 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 [9]. Those are the key qualities of ontology and it will be used in our research together with Enterprise Modelling.

II. The Background of the Research

Requirements Engineering (RE) is part of Systems engineering and has it's structure parts. It consists of requirements elicitation, analysis, evaluation, validation and management processes. The result of RE process is a document, software requirements specification (SRS). According to IEEE, SRS can be stated as a final result if it is: correct, unambiguous, complete, consistent, traceable, verifiable, extendable. From a user perspective, software requirements specification should be easy to read, written in understandable language for non-technical stakeholder and technical stakeholder, what means, there should not be any technological jargon. Also, it should be written in formally accepted template, have all the necessary parts included, be written in high quality language, with no grammar mistakes or similar. To sum up, after reading the system requirements specification, the responsible stakeholder should have a clear vision what system is developed for, what structural parts it will have, what workflow and problems it will solve.

The problem of this research is that, even many tools and methods been already presented in the industry, issues and difficulties still appear in requirements engineering. One of the difficulty is - quality of many specified requirements is poor. This means that far too many 'requirements' specified in real requirements specifications are ambiguous, not cohesive, incomplete, inconsistent, incorrect, out-of-date, specified using technical jargon rather than the terminology of the user or business/application domain, not restricted to externally-visible behaviour or properties of the system, infeasible to implement or manufacture, not actually mandatory (i.e., merely nice-to-haves on someone's wish list), irrelevant to the system being built, lacking in necessary metadata such as priority and status, untraced, in a form that is unusable to the requirements many stakeholders, unverifiable, and unvalidatable [7]. This problem most of the time can appear because of the lack of communication between stakeholders involved into requirements analysis process. Customer expressed requirements can be wrongly interpreted by the analyst and analyst can forward it wrongly to development team. And the process becomes like a chain of miscommunication. Reaching a common level of understanding of a problem domain is one of the key challenges that the software vendors and customers face during requirements definition. The process of articulating and clarifying business problems and arriving at a specification based on a shared understanding requires exchange and transfer of knowledge [25]. On the other hand, system analyst plays the key role to ensure the communication between the development team and the client. 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 too long [23]. The result of successful requirements engineering process vary on the experience and skills of the analyst. There should be a knowledge-based tool that overcome the lack of qualification of the analyst, which is not been presented to the area lately.

An ontology-based, Enterprise Metamodel supported requirements specification tool may help to reduce misunderstanding, missed information, and help to overcome some of the barriers that make successful acquisition of requirements so difficult.

III. Extended MOF architecture

The Meta-Object Facility (MOF) is an Object Management Group (OMG) standard. It is designed as a four-layered architecture, where the top layer is called the M3 layer. This M3-model is the language used by MOF to build meta-models, called M2-models. The most prominent example of a layer 2 MOF model is the UML meta-model, the model that describes the UML itself. These M2-models describe elements of the M1-layer, and thus M1-models, for example, models written in UML. The last layer is the M0-layer or data layer. It is used to describe real-world objects, instances [19]. See Fig. 1.

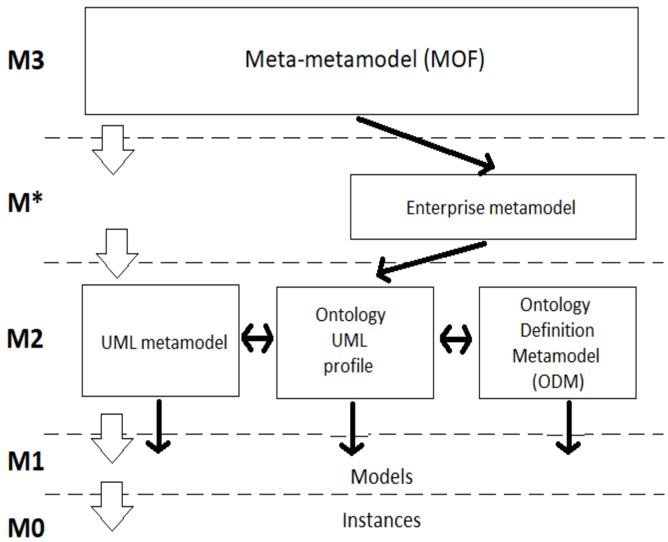

Fig 1. MOF architecture with additional Enterprise metamodel layer and Ontology modelling

Between M3 and M2 layers one more layer was added by [19], to assure more accurate usage of MOF architecture. This additional layer consists of Enterprise metamodel (EM). Enterprise metamodel is a formal structure which ensures more qualified information system development process and knowledge base data collection. Enterprise model and enterprise metamodel makes information system requirements models generation process more efficient and eligible and assure reduced number of mistakes in the final information system development phase [11].

The Ontology Definition Metamodel (ODM) is designed to include the common concepts of ontologies. In order to make use of the graphical modelling capabilities of UML, the ODM should have a corresponding UML profile [Sigel, 2001]. This profile will enable graphical editing of ontologies using UML diagrams as well as provide other benefits of using mature UML CASE tools. There is two-way mappings between: the ODM and the Ontology UML Profile and from the Ontology UML Profile to UML [5].

To structure domain knowledge, which is the key factor for successfully developed requirements specification, the methodology is needed. Requirements Engineering calls for an explicit domain knowledge. This domain knowledge generally resides in different areas, such as experiences, functionality, non-functional requirements, stakeholders and so on. Thus, it is necessary to concentrate this knowledge for the most appropriate application. Knowledge-driven techniques seem promising for this purpose. Kossmann et. al. in [15] define Knowledge-driven Requirements Engineering when Requirements Engineering is guided not only by a process but as well by knowledge about the process and the problem domain. In order to use knowledge-driven techniques, it is necessary to apply knowledge repositories that can be easily updated and utilised [14].

IV. Enterprise modelling and ontology 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 [10, 16]. Enterprise Modelling is an activity where an integrated and commonly shared model of an enterprise is created [9, 11, 22]. 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. For example, the Enterprise Model may contain processes modelled in BPMN, data modelled in ER and goals modelled in n*. The Enterprise Model is usually developed by several stakeholders, 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. Several Enterprise Modelling formal purposes are presented as follows [3, 17]:

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.

Enterprise Modelling (EM) aims to capture and represent organizational design in terms of business goals, processes, concepts, actors, as well as high level information system (IS) requirements by using conceptual models. Many EM techniques have emerged throughout the years, presenting different views of the enterprise and offering a wide variety of possibilities for designing, improving, re-structuring, and automating all or parts of the business in question [12; 22].

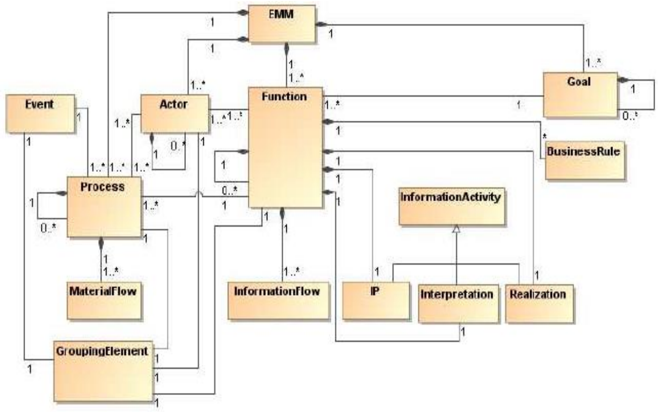

Fig. 2. The basic elements of Enterprise Metamodel (Reference: [4])

The researchers of Vilnius University and Kaunas University of Technology [4, 10, 22] developed a framework of Knowledge-based Enterprise model [Fig. 2], which helps to generate models that could be used for requirements specification. 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. Enterprise Model as organization's knowledge repository enables generate UML models with the help of transformation algorithms. Enterprise meta-model holds essential elements of business modelling methodologies and techniques, which ensures a proper UML models generation process [4, 22]. 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. It could be stated that this metamodel is part of MDA approach, this is why it is relevant to this research and it will be used in our framework [4].

Knowledge Based Subsystem, which improves traditional MDA conception with best practices of problem domain knowledge and user requirements acquisition methods, is presented in Fig 2 above. Usage of Enterprise Metamodel together with MDA improves the consistency of software artifacts and reduces IT projects dependency on empirical processes. The EM is intended to be formal structure and set of business rules aimed to integrate the domain knowledge for the IS engineering needs. It is used as the "normalized" knowledge architecture to control the process of construction of an EM [22].

EM mostly focuses on consistency of UML models generation. Also it is used as knowledge repository, where domain knowledge is stored. It's structure can be easily adapted to any domain, which means it is easily reusable. That is a huge advantage for the research. But even though, it has advantages, it has some drawbacks in a scope of requirements engineering too:

• It does not provide semantic concept of the requirements;

• It does not provide rules and logic for associations above requirements;

• It does not provide a shared common understanding of the structure of information among people or software agents;

• It does not provide rules for completeness, unambiguity and traceability criteria.

For the problems mentioned above solving, EM and ontology integration should be used. An ontology-based requirements specification tool may help to reduce misunderstanding, missed information, and help to overcome some of the barriers that make successful acquisition of requirements.

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 [18, 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 [6, 23].

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 [20, 23].

V. SYSTEM REQUIREMENTS SPECIFICATION GENERATION PROCESS

All of the system analysts would like to write requirements specifications meeting the criteria. But it is very connected with the experience of the system analyst, so human factor is playing a key role while preparing a specification. It also depends on the process how system requirements specification is developed. Unstructured and chaotic process leads to the failure or the result of the specification in the end will not meet the previously described criteria. The traditional requirements analysis process consists from these phases [13]:

• Requirements analysis;

• Functional analysis and allocation;

• System analysis and control;

• Design synthesis.

But this traditional process does not say anything about the sequence of the analysis, just the points that should be taken into account while designing requirements specification.

The quality of system requirements specification should be a repeatable process where competency questions written in natural language are interpreted. An algorithm by IEEE 830 standard is presented in the Fig. 3 below.

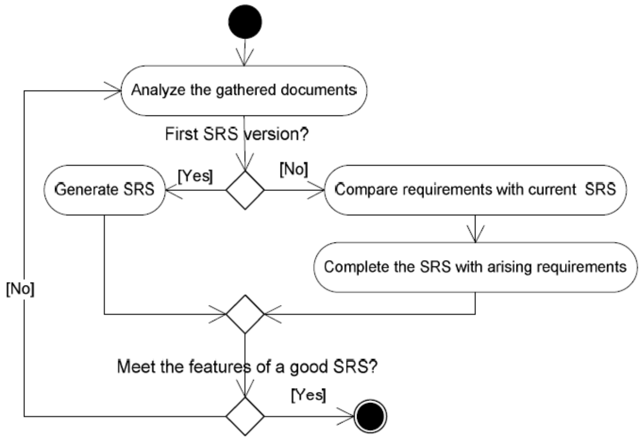

Fig. 3. SRS generation process (Reference: [13])

This process does not cover overall process of the analysis. For the process to be complete, we added domain node, as well as the summarized structure, also the rules executing and after all it goes to generating the final document of system requirements specification.

In the Figure 4, the upgraded top-level algorithm is presented. It gives the clarity of the process, also includes knowledge base and requirements ontology structure, as well as the rules to be executed for specification to meet the criteria. The very first step gives the base for exploring domain knowledge, the data for requirements design, basic statements of users expressed in natural language. After this analysis, the information should be summarized in a structural way, a requirements specification template should be presented. Requirements should be described in formal unified language to be easily understandable by development team. Also, it should be end-user friendly. If the result of the analysis satisfies the analyst, it can be described as a first system requirements specification version and then it can be moved to evaluating it according to criteria. But in a more complex way, after analysis, first version of the document is compared to the current system requirements specification. It means, it is compared to the knowledge already stored in a knowledge base and it is a process finding missing parts of the first version of system requirements specification.

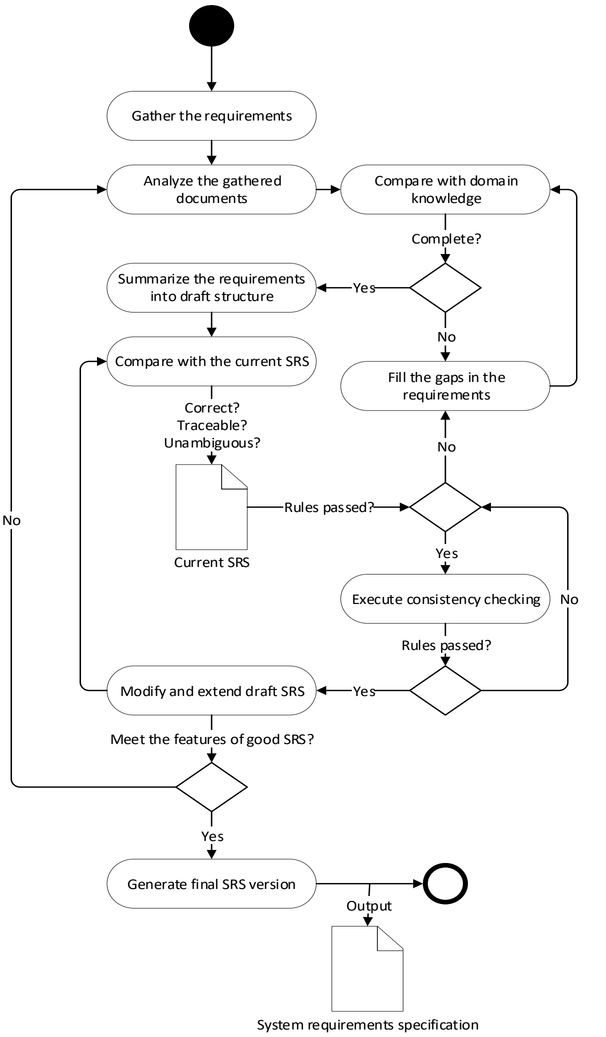

Fig. 4. Upgraded requirements analysis process

After comparing, specification is upgraded and completed. Then it can be evaluated to the criteria and if it meets the criteria, it can be stated as finished. If it does not meet criteria, analyst should start the process all over again, to polish the specification, because it cannot be outputted if it is not completed and the criteria hasn't been met. We will stick to this procedure when developing our method.

VI. The Metamodel of the Solution

Based on the MDA methodology and Ontology metamodel our solution was designed. It organizes knowledge among three contexts: Enterprise metamodel, Requirements Ontology and Requirements document template. The framework also incorporate abstractions from various knowledge modelling paradigms like feature models, business process models, data models and use case models, to capture and organize knowledge elements.

Below, there is a simple example showing the relationship between requirements, domain and the specification. It looks like domain stands on the root of the successful requirements specification project, but all of the three parts are essential and cannot be left behind [Fig. 5].

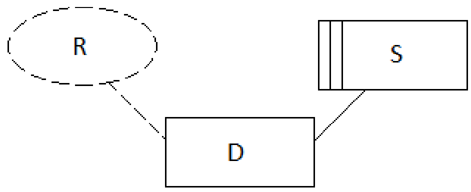

Fig. 5. The relationship between requirements, domain and specification (Reference: [24])

As illustrated in Figure 5, the role of requirements (R) in software engineering is to state relationships that are desired to hold between elements of a certain real world domain (D). Conversely, the role of a specification (S) is to provide instructions for a machine that has an interface to D so that the properties required in R hold. [24] Formally, this diagram can be explained as a logical relationship: $S \cup D \models R$. This diagram correlates to our solution, as there is the same three parts: Domain as Enterprise metamodel (EM), Requirements as Requirements ontology (RO) and Specification (S) as Requirements document template [Fig. 6].

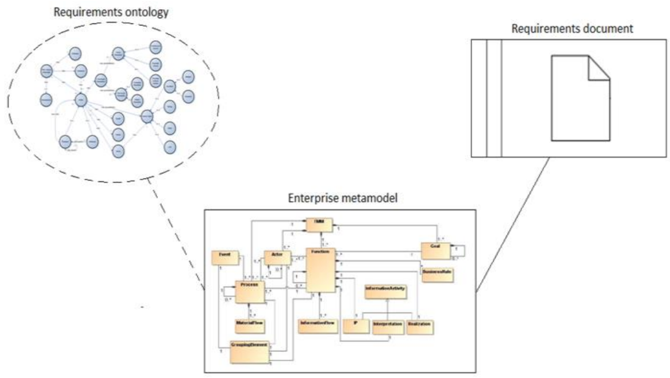

Fig. 6. The meta-metamodel of the solution

The complex structure of Enterprise metamodel, Requirements ontology and Requirements document template lets us to get overall vision about the requirements design and specification. It gives us knowledge base for domain and continuous improvement process for future projects, it also gives us structure of the requirements, it gives the clarity, effective analysis with the result of complete, consistent, unambiguous, extendable, modifiable, traceable and correct requirements specification. Association between Enterprise metamodel and Requirements ontology is realized through the transformation algorithms.

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 [22, 23]. By giving us structure not specified by a concrete domain, Enteprise Metamodel ensures reusability, modifiability and flexibility to the method.

VII. CONCLUSIONS AND FUTURE WORKS

An ontology with EM-based requirements specification tool may help to reduce misunderstanding, missed information, and help to overcome some of the barriers that make successful acquisition of requirements so difficult. Key argument why additional solution is needed is that in existing ones requirement knowledge is not sufficiently covered. Intentions, risks, obstacles and decisions are not documented during RE and thus, are not available at later stages during software development.

For domain knowledge repository, MDA based EM was chosen as the most relevant approach as it stands out for classic methodologies. So combined MDA and ODM methodologies, we can get great results. Using ontologies with Enterprise Modelling offers several advantages. Ontologies ensure clarity, consistency, and structure to a model. They promote efficient model definition and analysis.

In the future works it is planned to continue the research in the area and present the deeper vision about the proposed method. To create the full process of upgraded requirements engineering process with incorporated Enterprise Metamodel (EM) and Requirements Ontology (RO) in the process. To create and present mappings between EM and RO notations, also the transformation rules. Also it is planned to present the whole structure of the method by adding good requirements specification criteria by IEEE as a base, to show that the method stands out for the remaining problems in requirements engineering.

VIII. REFERENCES

[1] Alonso Julita B. (2006). Ontology-based Software Engineering, Engineering Sup-port for Autonomous Systems. ASLab-ICEA-R-2006-016 v 0.1 Draft of 2006-11-15.

[2] Bera, P., Burton-Jones, A., Wand, Y. (2011) Guidelines for Designing Visual Ontologies to Support Knowledge Identification. Mis Quarterly 35, 883–908.

[3] Brathaug T. A.&Evjen T.Å. (1996) Enterprise Modelling. SINTEF, Trondheim.

[4] Butleris, R., Lopata, A., Ambraziunas, M., Veitaitė, I., Masteika S., SysML and UML models usage in knowledge based MDA process. Elektronika ir elektrotechnika. Vol 21, No 2 (2015) pp. 50-57 Print ISSN: 1392-1215, Online ISSN: 2029-5731.

[5] Dragan Djuric. Dragan Gasevic, Vladan Devedzi (2006) Model Driven Architecture and Ontology Development. ISBN-10 3-540-32180-2 Springer Berlin Heidelberg New York.

[6] Fadel, G., Fox, M., Gruninger, M. (1994). A Generic Enterprise Resource Ontology. In: Proceedings of the 3rd Workshop on Enabling Technologies: Infrastructure for Collaborative Enterprises. p. 117-128.

[7] Firesmith D. G. (2003) Engineering Security Requirements. in Journal of Object Technology, vol. 2, no. 1, January-February 2003, pages 53-68.

[8] Force, S. E. T. (2001). Ontology driven architectures and potential uses of the semantic web in systems and software engineering.

[9] Frederik Gailly, Sven Casteleyn, Ndejda Alkhaldi (2013) On the Symbiosis be-tween Enterprise Modelling and Ontology Engineering. Ghent University, Universi-tat Jaume I, Vrije Universiteit Brussel, DOI: 10.1007/978-3-642-41924-9_42.

[10] Gudas S. Enterprise knowledge modelling: Domains and aspects. Technological and economic development of Economy. Baltic Journal on Sustainability 281–293, 2009.

[11] Gudas S., Architecture of Knowledge-Based Enterprise Management Systems: a Control View, Proceedings of the 13th world multiconference on systemics, cybernetics and informatics (WMSCI2009),) July 10 – 13, 2009, Orlando, Florida, USA, Vol. III, p.161-266 ISBN-10:1-9934272-61-2 (Volume III).

[12] Iyad Zikra, Janis Stirna, Jelena Zdravkovic (2017) Bringing Enterprise Modelling Closer to Model-Driven Development. Oslo, Norway. Springer, Lecture Notes in Business Information Processing, LNBIP-092, pp.268-282.

[13] IEEE 830 standard.

[14] Katja Siegemund, Edward J. Thomas, Yuting Zhao, and Uwe Assmann (2010) Towards Ontology-driven Requirements Engineering.

[15] M. Kossmann, R. Wong, M. Odeh, and A. Gillies. Ontology-driven Requirements Engineering: Building the OntoREM Meta Model. In Information and Communication Technologies: From Theory to Applications, 2008. ICTTA 2008. 3rd International Conference on, pages 1 – 6, 2008.

[16] Mark S. Fox, Mihai Barbuceanu, Michael Gruninger, and Jinxin Lin (1998), An Organization Ontology for Enterprise Modelling. MIT Press Cambridge, MA, USA p. 131-152, ISBN:0-262-66108-X.

[17] Nadeem Ahmed Khan (2011) Transformation of Enterprise Model to Enterprise Ontology. Master Thesis, Jonkoping, Sweden.

[18] OMG ODM (2014) OMG Formal Versions of Ontology Definition Metamod-el// http://www.omg.org/spec/ODM/1.1

[19] OMG MOF., (2015) OMG Meta Object Facility (MOF) Core Specification. Version 2.4.2. April 2014, http://www.omg.org/spec/MOF/2.4.2

[20] Ostie James K. (1996). An Introduction to Enterprise Modelling and Simulation.

[21] Perjons, E (2011) Model-Driven Process Design. Aligning Value Networks, Enter-prise Goals, Services and IT Systems. Department of Computer and Systems Sci-ences, Stockholm University. Sweden by US-AB, Stockholm ISBN 978-91-7447-249-3.

[22] Veitaite I., Ambraziunas M., Lopata A. (2014) Enterprise Model and ISO Stand-ards Based

Informations System's Development Process. 16th International Con-ference on Business Information Systems, BIS2014 International Workshop, Larnaca, Cyprus, May 22-23, 2014, Series: Lecture Notes in Business Infor-mation Processing.

[23] Veitaitė I., Lopata A., Žemaitytė N. (2016) Enterprise Model based UML Interaction Overview Model Generation Proces. 19th International Conference on Business Information Systems, BIS2019 International Workshop, Series: Lecture Notes in Business Information Processing. ISBN 978-3-319-26762-3.

[24] Zoghwi D., Gervasi V. The three Sc of requirements: consistency, completeness and correctness.

[25] Ghaisas S., Ajmeri N. (2013) Knowledge-Assisted Ontology-Based Requirements Evolution. In: Maalej W., Thurimella A. (eds) Managing Requirements Knowledge. Springer, Berlin, Heidelberg.