Baltic J. Modern Computing, Vol. 7 (2019), No. 2, 190-210
https://doi.org/10.22364/bjmc.2019.7.2.02

Ontology and Enterprise Modelling Driven Software Requirements Development Approach

Neringa MAKRICKIENĖ[1], Saulius GUDAS[1], Audrius LOPATA[2]

[1]Vilnius university, Kaunas faculty, Muitinės str. 8, Kaunas 44280, Lithuania
[2]Kaunas university of technology, Studentų str. 48, Kaunas 51367, Lithuania

neringa.makrickiene@gmail.com,saulius.gudas@mii.vu.lt,
audrius.lopata@knf.vu.lt

Abstract. The aim of the research is to analyze latest methodologies in requirements engineering process, the newest proposals to upgrade the process and to present an approach helping to address challenges in the area, such as lack of quality of requirements specifications and requirements due to system analyst skills and experience. The areas of the research includes ontology engineering practices, as well as part of Model Driven Architecture (MDA) approach, to be more specific – Enterprise Metamodel (EMM). This paper presents Metamodel for knowledge-based subsystem merged with requirements ontology and requirements specification template to cover all the aspects in problematic requirements engineering process.

Keywords: Requirements engineering, Enterprise Metamodel (EMM), Enterprise Model (EM), Model Driven Architecture (MDA), Ontology, software requirements specification (SRS).

1. Introduction

Requirements Engineering (RE) is crucial part of Systems Engineering (SE). It is widely acknowledged amongst industry practitioners and researchers that software development projects are critically vulnerable when the requirements related activities are performed poorly. Therefore, requirements engineering 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 (Alonso, 2006; Makrickienė et al., 2018).

Many tools and methods have been presented in the requirements industry already, but issues and difficulties appear: quality of many specified requirements is still poor. This means that far too many 'requirements' specified in real requirements specifications are ambiguous, incomplete, inconsistent, incorrect, out-of-date, non-verifiable, non-validatable, sometimes it is specified using technical jargon rather than the terminology mutually understandable for the user and the development team. Also, many requirements are not describing exact behaviour or properties of the system, impossible to implement, lacking in necessary metadata (e.g. priority, status etc.) or just unusable to

the stakeholders (Firesmith, 2003). These problems can appear because of the lack of communication between stakeholders involved into requirements analysis process. 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 (Ghaisas et al., 2012; Siegemund et al., 2010).

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 so difficult (Firesmith, 2003).

The paper is structured as follows: in the second section, some background information about the problems arising in requirements engineering is provided, our motivation into it, as well as, comparison of existing methodologies in the area. In the third section, we will describe our solution. In the fourth section, we will provide a case study to demonstrate the solution and finally, in the fifth section of the article, conclusions and future works are presented.

In this article, a comparative analysis of existing methodologies was made, conceptual schema of the method is presented, a case study identifying the elements of the two systems user stories is presented, and also, a mapping of requirements ontology and enterprise model is made.

2. Related work

In this section related work done by other authors will be presented. We will consider only upper level (top-level) ontologies as it is referring to metamodel approach.

Some of ontologies, like OntoREM (Kossmann et al., 2008), CORE (Jureta et al., 2009), SWORE (Riechert et al., 2007) made itself as a brands and were successfully integrated as a great tools helping to design requirements in various projects. While others like, i*/Tropos (Castro et al., 2002), OntoSRS (Castañeda et al., 2010), KAOS (Dardenne et al., 1993), Farefelder et al. (2011), Kof (2004) are only mentioned in various scientific articles and researches.

While domain knowledge use in requirements engineering stands as a base for our research, it would be important to mention similar approaches. OntoSRS (Castañeda et al., 2018), OntoREM (Kossmann et al., 2008), FRS Ontology (Haruhiko et al., 2005), SWORE (Riechert et al., 2007) incorporated domain knowledge as a base for developing requirements. OntoREM (Kossmann et al., 2005) is a comprehensive specification of the ODRE methodology, including the underlying concepts in the RE domain and relationships between them. It consists of the OntoREM metamodel ontology and a number of domain ontologies. FRS Ontology (Haruhiko et al., 2005) method is based on a domain ontology and a mapping to the requirement specification. SWORE (Riechert et al., 2007) supports the RE process semantically, provides a semantic structure for capturing requirements information and linking this information to domain and application specific vocabulary. OntoSRS (Castañeda et al., 2010) framework is divided into three application areas, such as: the description of requirements specification documents, the formal representation of the application domain knowledge, and the formal representation of requirements.

NFR Ontology (Sancho et al., 2007) considers mainly non-functional requirements, while FRS Ontology (Haruhiko et al., 2005) method is opposite - evaluates only functional requirements.

Some of the methods greatly explored one of the problems, such as requirements inconsistency. Ying et al. (2008) proposed a method which detects and resolves inconsistencies of domain ontologies for requirements engineering. That is checking whether the ontology is satisfiable, which means that there is no contradicting information in the ontology. Meanwhile, Zhu et al. in (2005) proposed ontology-based approach for inconsistency measurement of requirements specifications based on a requirements refinement tree. Therefore, requirements are stepwise decomposed until a requirement can be realized.

Farefelder et al. (2011) and Kof (2004) methods also use ontologies in requirements engineering process, mostly focusing on requirements elicitation. Both methods provide tools how to work on semi-formal requirements expressed in natural, human readable language. Such approach is very important for the analyst to better understand and interpret requirements to computer readable languages (Siegemund et al., 2010).

These described methods are all different, yet they have some similarities. To compare them, formal criteria should be noted. To compare all of these methods that use ontologies to solve various requirements engineering problems, IEEE 830 standard criteria (software requirements should be correct, consistent, unambiguous, complete, extendable, modifiable, traceable) was taken into account, as each of these methods express some of the criteria.

Table 1. Existing solutions compared

Criteria/ Methods	Correctness	Consistency	Unambiguity	Completeness	Extendability	Modifiable	Traceable
CORE	-	+	+	-	+	+	-
ontoREM (ODRE)	-	+	-	+	-	-	+
i*/Tropos	-	+	-	-	-	-	-
FRS	-	+	-	+	-	-	+
SWORE	+	-	-	+	+	+	-
NL approach	-	-	-	-	+	+	-
Req.refinement tree	+	-	-	+	-	-	-
NFR (non-func.)	-	-	+	-	-	-	-
Inconsistency checking	-	+	-	-	-	+	-
KAOS	-	-	-	-	+	+	-
K-RE			+			+	

During this analysis, conclusions were made, that already proposed methods do not cover:

- 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.

- Most of the solutions do not meet correctness and traceability requirements. Also very few cover unambiguity and completeness criteria.
- Requirement problems (e.g. conflicts, unstated information) are detected too late or not all.
- To dig deeper into realisation of the methods, relationships among requirements are inadequately captured and are often limited to binary relations between requirements instead of defining which kind of relation is meant (e.g. excluding, alternative, generalization).
- Methods need richer and higher-level abstractions.
- Some of the methods are goal-oriented on requirements engineering, so that means it does not cover domain knowledge, scenario-based requirements.
- Some of the methods are incomplete or oriented only to functional or non-functional requirements.
- Not all of the methods and tools are still supported, which shows that they were not very successful and beneficial.
- Just a few methods are oriented to requirements analysis in the RE process. Mostly of them are oriented to requirements elicitation.
- Also, one of the main problems which is very relevant to our research is that not all of the methods listed above are based on MDA architecture. ODRE implicates or mentions ISO standards base of the requirements engineering while developing the method. OntoSRS gives brief reasoning of the method based on IEEE 830 standard.

Although many approaches for representing and applying ontologies in requirements engineering have already been devised, they haven't found their way into enterprise applications (Siegemund et al., 2010). Summarizing literature study on the existing solutions, there is no proposal that covers all criteria of the good requirements specification. Based on the existing methodologies analysis, it was concluded, that even there are promising approaches proposed, but it still lacks overall point of view to the main problems of requirements engineering. So, it is beneficial to propose **new method** to solve these problems.

3. The solution

3.1. The Meta-metamodel

The solution was designed by merging promising technologies in Requirements Engineering. Enterprise Metamodel filled in with data and transformed to Enterprise Model stands for knowledge structure, to be more specific, knowledge base for the requirements for the specific domain. Requirements Ontology stands for the requirements structure, semantic validation of the requirements, as well as knowledge capture by overlapping Enterprise Model in some areas. And also, requirements specification documents stands as a result of capturing requirements, it's structure followed by standards, should be defined by following quality criteria of good requirements specification: consistency, unambiguity, completeness, traceability, modifiability, extendability.

The complex structure of Enterprise Metamodel, Requirements ontology and Requirements document template lets us to get overall vision about the requirements design and specification (Makrickienė et al., 2018). 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, by avoiding missinterpretations, ensure semi-automated software requirements specification generation process. Association between Enterprise Metamodel and Requirements ontology is realized through the transformation algorithms.

Since ODM and MDA are matching technologies, it is useful to use both of these technologies together to improve requirements engineering processes. It will provide a good support in software tools and ease the integration with existing or upcoming software tools and applications, which will add values to both sides (OMG MOF, 2013). MDA and its four-layer architecture provide a solid basis for defining metamodels of any modeling language, so it is the straight choice to define an ontology-modeling language in MOF.

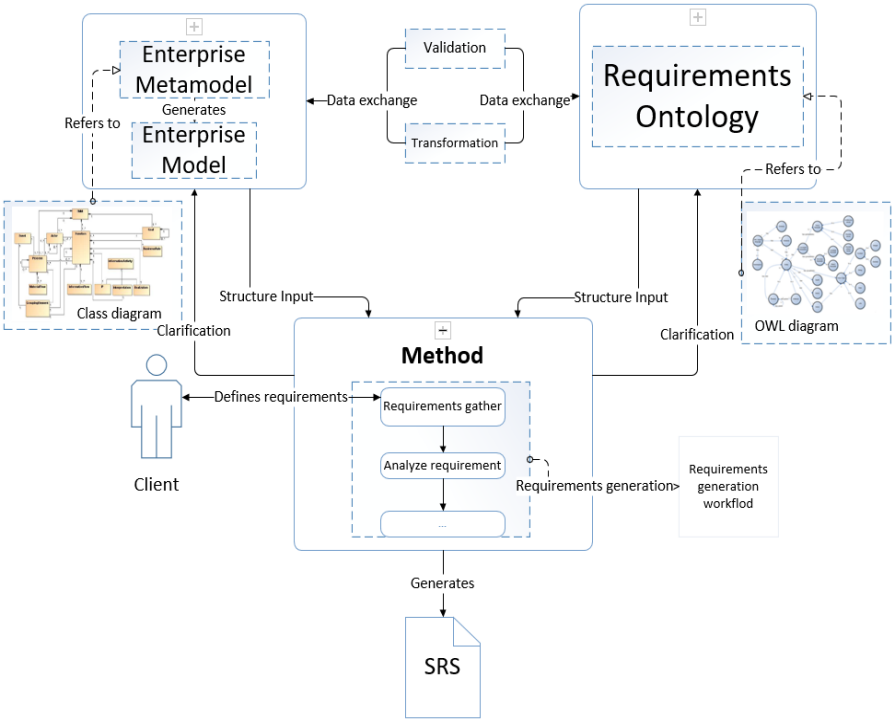


Fig. 1. Conceptual schema of the method

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 incorporates abstractions from various knowledge modeling paradigms like feature models, business process models, data models and use case models, to capture and organize knowledge elements.

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 (Lopata et al., 2012; Veitaitė et al., 2016). By giving a structure not specified by a concrete domain, Enteprise Metamodel ensures reusability, modifiability and flexibility to the method.

Summarizing the elements, merging it together to the structure, conceptual schema of the method is presented in the figure 1 above. The method itself, lets the requirements expressed by the potential user to be summarized by the system analyst by using the method. It helps to overcome lack of experience and skills of the professional who captures and specifies requirements of the system in some specific domain area. It gives the requirements engineering process support tool for specifying the requirements. It helps to generate software requirements specification meeting the IEEE criteria (IEEE 830 standard), by using domain knowledge and requirements relations in the ontology support.

The solution is based on an Enterprise Metamodel and Requirements Ontology repository. This repository serves as knowledge base, providing a sufficient structure to capture all relevant requirements artefacts and predefines a set of meaningful metadata to formalize requirements knowledge and allow the validation of quality criteria. Additionally, the solution provides a workflow and structure to automatically generate a requirements specification regarding a set of predefined attributes the System Analyst can select from.

The method is used for the research purposes, but it can be expanded as a plug-in integrated into the CASE (*Computer Aided Software Engineering*) tool for requirements specification design and generation. It is designed as a knowledge-based subsystem as CASE tool component with Enterprise Metamodel and requirements ontology inside.

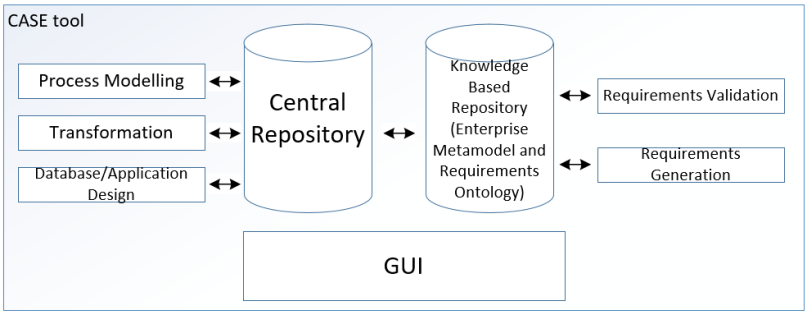


Fig. 2. Components of Knowledge-based CASE tool

In a knowledge-based computerized IS engineering, project models can be generated interactively by using generation algorithms, if the necessary knowledge will be collected into knowledge repository. Knowledge in the repository is verified to ensure

automatically generated design models and software code quality in knowledge gathering into knowledge repository phase.

3.2. Enterprise Metamodel

While analysing the area of interest and related methodologies, we summarised several problematic key points that have to be taken into account while developing the method.

A framework of Knowledge-based Enterprise model, which helps to generate models, that could be used for requirements specification is presented in (Gudas, 2009; Lopata, 2004; Lopata et al., 2012).

Knowledge-based CASE systems holding substantial components, which organize knowledge: knowledge-bsed subsystem's knowledge base, which essential elements are Enterprise Metamodel specification and Enterprise Model for certain problem domain (Gudas, 2009; Lopata, 2004; Lopata et al., 2012). Enterprise Model as organization's knowledge repository enables generate UML models with the help of transformation algorithms. Enterprise Metamodel specifies essential elements of business modelling methodologies and techniques, which ensures a proper UML models generation process (Lopata et al., 2012; Morkevicius et al., 2011). 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 (Veitaitė et al., 2016).

3.3. Requirements Ontology

Ontologies have long been used in the knowledge engineering community to perform conceptual domain modelling. In this domain, ontologies are interpreted as "an explicit specification of a shared conceptualisation" (W3C OWL). Ontologies contain explicitly defined and generally understood concepts and constraints that are machine understandable. More broadly, ontology specifies concepts that directly relate to being (from philosophical point of view), in particular becoming, existence, reality, as well as the basic categories of being and their relations (Siegemund et al., 2010). These definition are the key reasons why ontologies are continuously been used in requirements engineering area. Different authors (Siegemund et al., 2010; Kossmann et al., 2008; Castañeda et al., 2010; Firesmith, 2003, W3C OWL) agree, that ontology engineering is a subfield of Knowledge Engineering and concerned with methods and methodologies for building ontologies (Siegemund et al., 2010). For this reason, ontology can be cooperated with an Enterprise Metamodel to capture the whole picture of knowledge and domain for requirements design.

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 (Fadel et al., 1994).

In order to address the problem area, requirements ontology was created. Formal templates are used for designing system requirements specifications, such as:

- Volere Requirements Specification Template, copyright © 1995 – 2018 the Atlantic Systems Guild Limited;
- IEEE template 830;
- IBM template;
- ISO standard (ISO/IEC 25001:2014) and others.

In this article, the fragment of the Requirements Ontology will be presented. As the base of this ontology, Volere requirements specification template key objects were taken and they stand as top entities. This ontology represents all critical parts of requirements specification, but also it can be easily extended and upgraded if needed, also as it's structure will correlate with several standard templates, it can generate requirements specification into several different structures, based on these templates. The main entities of the fragment structure are: Introduction, Overall description, Requirements.

Requirements Ontology was created using open source Protégé 4.3 tool, by giving it entities, relations between entities, also properties about each entity. In the figure 2 below, top entities and the hierarchical structure of the Requirements Ontology is presented.

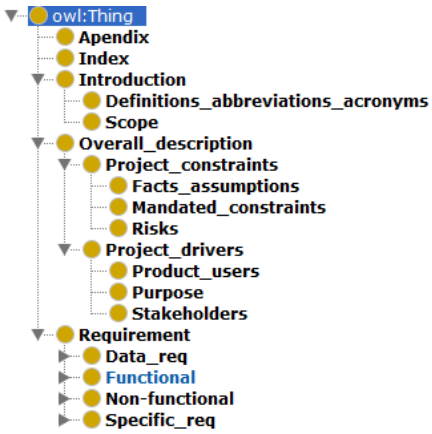


Fig. 3. Top level entities of Requirements Ontology

In order to identify relationships between Requirements Ontology entities, relations table is presented below.

Identified relations among Requirements Ontology classes are visualized in a class diagram and presented in the figure below. Visualization was made using MS Visio tool.

In the Requirements Ontology structure, additional classes, sub-classes and relations can be added accordingly in demand. This option gives the method ability to adapt to changing requirements and makes it adaptable knowledge based process.

Table 2. Relations table of the Requirements Ontology

Class name	Relation	Cardinality	Sub-class name	Inverse relation
ReqOntology	ReqOntoHas	1	Introduction	isPartOf
ReqOntology	ReqOntoHas	1	Overall description	isPartOf
ReqOntology	ReqOntoHas	1..*	Requirements	isPartOf
ReqOntology	ReqOntoHas	1	Index	isPartOf
ReqOntology	ReqOntoHas	0..1	Apendix	isPartOf
Introduction	IntroductionHas	1	Definition, abbreviations, acronyms	isPartOf
Introduction	IntroductionHas	1	Scope	isPartOf
Overall description	OverallDescriptionHas	1..*	Project drivers	isPartOf
Overall description	OverallDescriptionHas	1..*	Project constraints	isPartOf
Requirements	RequirementsHas	1..*	Data requirements	isPartOf
Requirements	RequirementsHas	1..*	Functional requirements	isPartOf
Requirements	RequirementsHas	1..*	Non-functional requirements	isPartOf
Requirements	RequirementsHas	0..1	Specific requirements	isPartOf

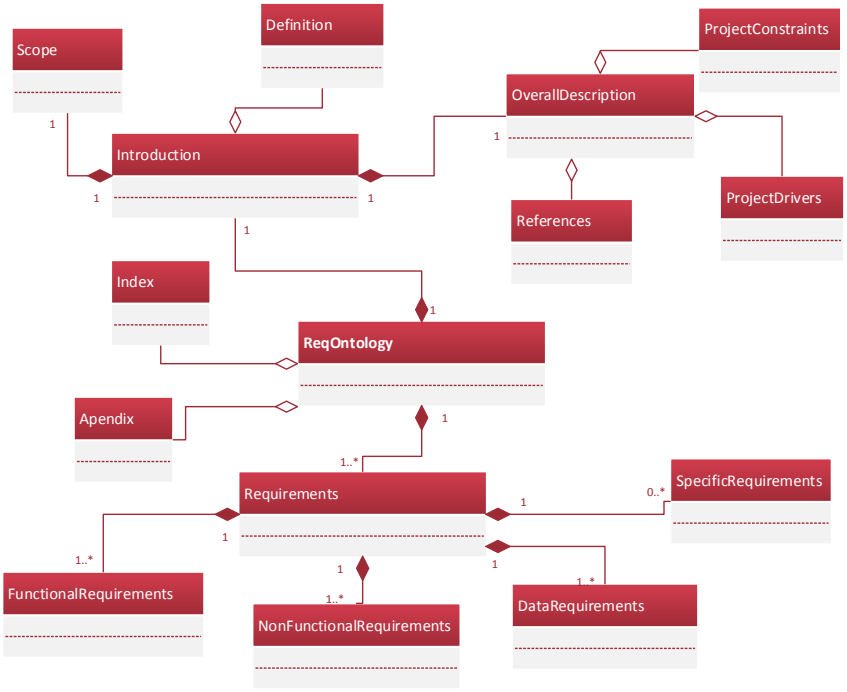


Fig. 4. Requirements Ontology (UML class diagram)

3.4. Relationship between Requirements Ontology and Enterprise Metamodel

Alonso (2006) describes ontology as a set of rules. In order to validate ontology within domain knowledge and business processes, it is extendend with Enteprise Metamodel. Enteprise Metamodel conceptual schema was presented in (Lopata, 2004; Makrickienė et al., 2018). Class diagrams will be provided to have the same format of Requirements Ontology and Enterprise Metamodel. Relations of the original Enterprise Metamodel were identified in the table below.

Table 3. Relations table of the Enterprise Metamodel

Class name	Relation	Cardina-lity	Sub-class name	Inverse relation
EMM	EMMHas	1..*	Process	isPartOf
EMM	EMMHas	1..*	Function	isPartOf
EMM	EMMHas	1..*	Goal	isPartOf
EMM	EMMHas	1..*	Actor	isPartOf
Event	EventHas	1	Process	isPartOf
Process	ProcessHas	1..*	MaterialFlow	isPartOf
MaterialFlow	MaterialFlowHas	1	MaterialInputFlow	isPartOf
MaterialFlow	MaterialFlowHas	1	MaterialOutputFlow	isPartOf
Actor	ActorHas	1	ProcessActor	isPartOf
Actor	ActorHas	1	FunctionActor	isPartOf
Function	FunctioHas	1	Process	isPartOf
Function	FunctioHas	1..*	InformationFlow	isPartOf
Function	FunctioHas	1	InformationProcessing	isPartOf
Function	FunctioHas	1..*	Realization	isPartOf
Function	FunctioHas	1..*	InformationActivity	isPartOf
Function	FunctioHas	1..*	BusinessRules	isPartOf
InformationFlow	InformationFlowHas	1..*	ProcessOutput	isPartOf
InformationFlow	InformationFlowHas	1	InformationProcessingInput	isPartOf
InformationFlow	InformationFlowHas	1	InformationProcessingOutput	isPartOf
InformationFlow	InformationFlowHas	1	ProcessInput	isPartOf
InformationActivity	InformationActivityHas	1	InformationProcessing	isPartOf
InformationActivity	InformationActivityHas	0..1	Interpretation	isPartOf
BusinessRules	BusinessRulesHas	0..1	BRInterpretation	isPartOf
BusinessRules	BusinessRulesHas	0..1	BRRealization	isPartOf
BusinessRules	BusinessRulesHas	1	BRInformationProcessing	isPartOf

Identified relations among Enterprise Metamodel classes, are visualized in a class diagram and presented in the Fig. 5.

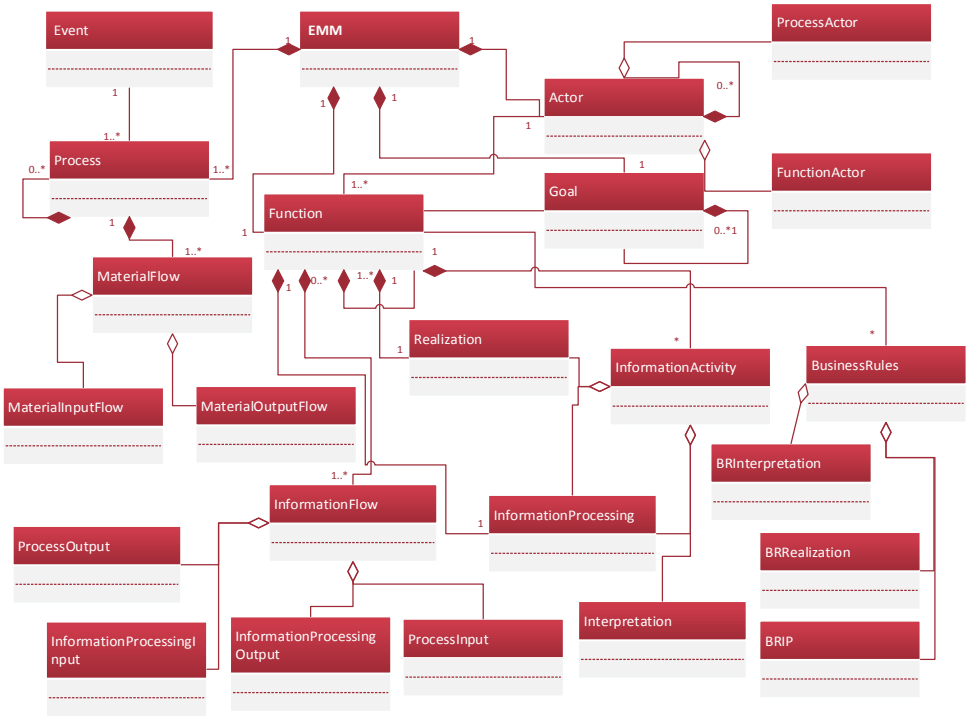
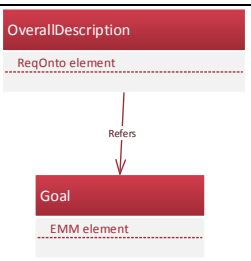
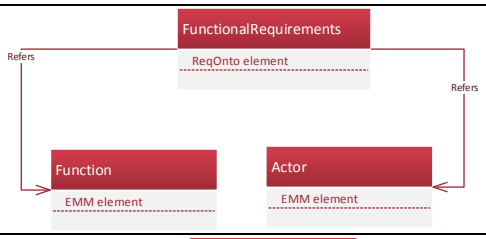
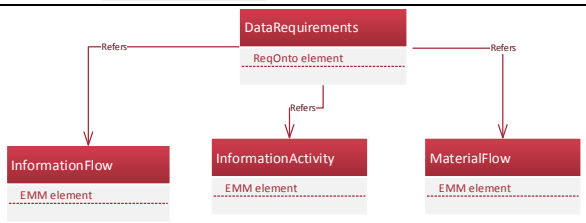
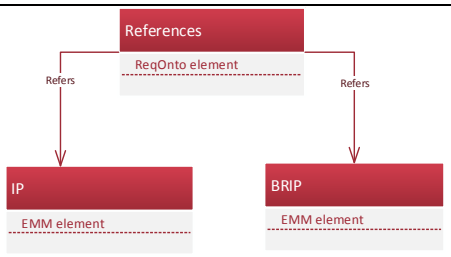


Fig. 5. Enterprise Metamodel (UML class diagram)

Below a class diagram of the Enteprise Metamodel and Requirements Ontology mapping is presented. Overall schema will be divided into parts to be more readable. From the map we can extract main Classes that have connections between Enterprise Ontology and Requirements Ontology.

The mapping between Requirements Ontology and Enterprise Metamodel shows that these two approaches are well compatible when expressed in the common format and level, elements have relations among each other as requirements refer to enterprise domain knowledge elements and supplement each other. Lower layers of Requirements Ontology and Enterprise Metamodel already have transformations between each other, as Requirements Ontology is expressed in OWL language and Enterprise Metamodel is expressed in UML. Furthermore, we will put these schemas into real world case and see if this mapping works in real life cases.

Table 4. Mapping of Enterprise Metamodel and Requirements Ontology

Description	Schema
Overall Description (Requirements Ontology) refers to Goal (EMM)	
Functional Requirements (Requirements Ontology) refers to Actors (EMM) and Function (EMM)	
Data Requirements (Requirements Ontology) refers to Information Flow (EMM), Information Activity (EMM) and Material Flow (EMM)	
References (Requirements Ontology) refers to InformationProcessing (EMM) and BRInformationProcessing (EMM)	

4. Case study

The case study will be organized as a process. Workflow is presented in Fig. 6..

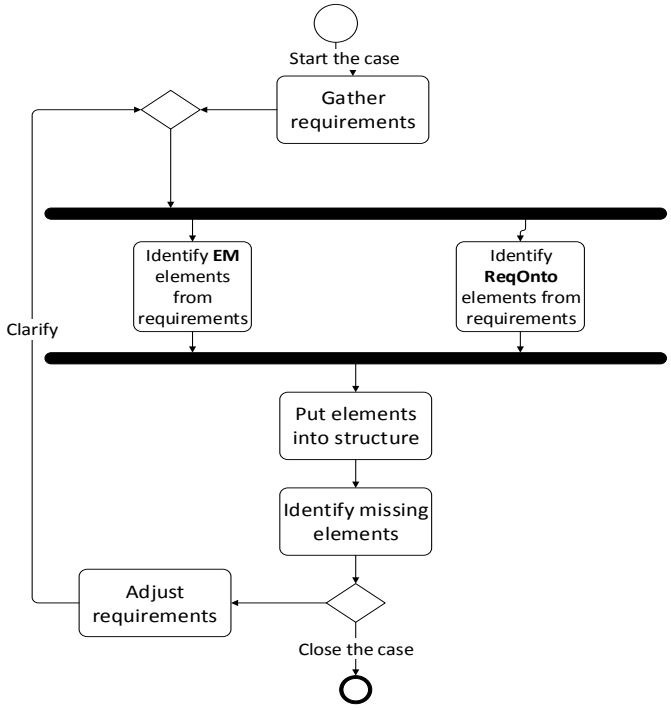


Fig 6. Case study workflow

Two real case scenarios will be presented in the eksperiment and compared. First real case scenario was chosen to be related to e-commerce, order processing process and is taken from the (Lopata et al., 2014), where it was generated by using Enterprise Metamodel approach.

For the real case scenario we have user requirements, expressed in natural language as 10 user stories.

Table 5. User stories – order management system

Number	Statement
1.	As a client, I request an order of goods.
2.	As a manager, I receive order from the client.
3.	As a manager, I fill the order form.
4.	As a manager, I send an invoice to the client.
5.	As a client, I accept the invoice.
6.	As a client, I make a payment.
7.	As a manager, I accept the payment.
8.	As a manager, I ship the order.
9.	As a client, I accept the order.
10.	As a manager, I close the order case.

The activity diagram was chosen to visualize the case (Lopata et al., 2014). In our approach, this case will address different problematics, the elements of the Enterprise Metamodel and will be incorporated in Enterprise Metamodel and Requirements Ontology structure for checking if the method can be validated with this real world case scenario.

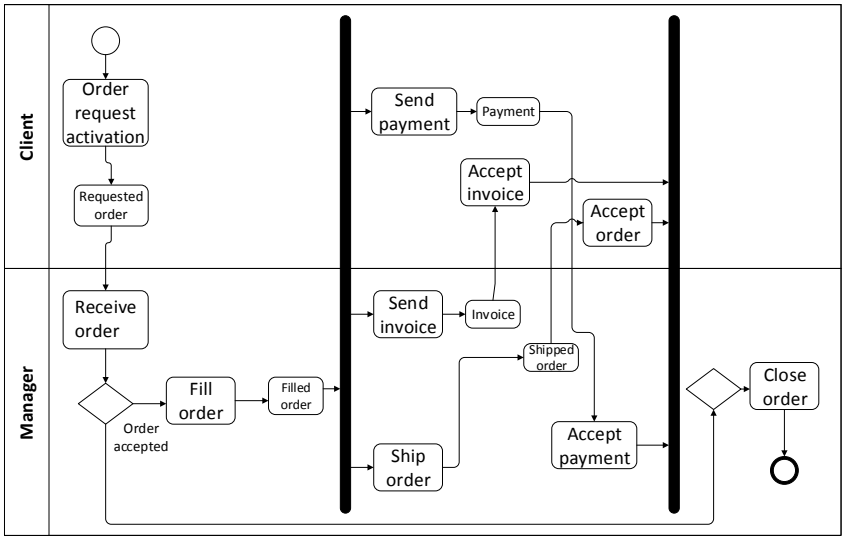


Fig. 7. Activity diagram of order management system

First of all, order processing elements in Enteprise Metamodel structure were identified in (Lopata et al., 2014), and the next step is to identify it in Requirements Ontology structure. After that, we will put these order processing elements in the Enterprise Metamodel and Requirements Ontology map to have overall vision of the case study.

Table 6 represents the example in which enterprise model elements are reflected as UML activity diagram model. Business activity of order request and its closure is used as example for UML activity diagram generation process. Order requested activation is starting elements from client side and requested order is input parameter of the activity. After the order is accepted and all the required information is filled in, the payment is accepted and the order is shipped. Note, that this business flow allows order shipment before the invoice is sent or the payment is confirmed (Lopata et al., 2014). Enterprise Model elements identify domain elements, that needs to be included in requirements specification to have fully described real world case. The structure of the Enterprise Model lets us to see what elements are described, what elements missing in order to have complete requirements specification.

Once Enterprise Model elements are identified, Requirements Ontology elements has to be identified and defined. As mentioned above, we have 10 user stories. From this information, Requirements Ontology elements can be extracted.

Table 6. Enterprise Model elements – order management system

EM element \ Real case element		Manager	Client	Initial Node	Order Request Activation	Requested Order	Receive Order	Decision Node	Fill Order	Filled Order	Fork Node	Send Invoice	Send Payment	Payment	Invoice	Ship Order	Shipped Order	Accept Payment	Accept Invoice	Accept Order	Join Node	Merge Node	Close Order	Activity Final Node
Actor		+	+																					
Event					+		+		+			+	+			+		+	+	+				+	
Process	Material Input				+							+	+	+	+	+	+							
	Material Output					+				+				+	+		+	+	+	+				
Function	Business Rules			+				+			+										+	+		+
	Informa-tion Flow					+				+														
	Informat ion Activity				+		+		+														+	

Table 7. Requirements Ontology elements – order management system

Real case element	**Req Ontology element**
A new order management system.	Definition
Statements: New platform is created; Integration with payment gateway needed; Interaction among 2 actors, 5 data elements; 10 task elements and 6 process elements.	Scope
Statements: • Client must have possibility to request an order. • Manager must have possibility to receive order from the client. • Manager must have possibility to fill order. • Manager must have possibility to send an invoice. • Client must have possibility to accept invoice. • Client must have possibility to send payment. • Manager must have possibility to accept payment. • Manager must have possibility ship order to client. • Client must have possibility to accept order. • Manager must have possibility to close the order.	Functional requirements
Statements: • System must be reachable for both via internet: client and manager interaction; • Integration with payment should be provided • System must be easy to use.	Non-functional requirements
• 2 actors: Manager; Client. • 5 data elements: Requested order; Filled order; Invoice; Payment; Shipped order. • 10 task elements: Order request activation; Receive order; Fill order; Send invoice; Accept invoice; Send payment; Accept payment; Ship order; Accept order; Close order. • 6 process elements: Initial Node; Decision Node; Fork Node; Join Node; Merge Node; Activity Final Node.	Data requirements

Not all of the elements were identified and covered. It is good, because during such process we can see what elements are missing, what elements have to be eliminated as not mandatory for the specific case and what elements needs to be adjusted to fulfill the requirements domain.

To make sure, this situation is affected not only because of the case data, but in general, we will do a case study with additional case. The second real case scenario was chosen to be similar to e-commerce, but with different elements and workflow – purchase requisition review and approval management system designed by Dynamics365.

Table 8. User stories – purchase management system

Number	Statement
1.	As a requester, I request a purchase requisition.
2.	As a purchasing agent, I receive purchase requisition.
3.	As a purchasing agent, I review purchase requisition.
4.	As a purchasing agent, I update purchase requisition.
5.	As a purchasing agent, I can approve purchase requisition automatically.
6.	As a purchasing agent, I can send for approval to requester's manager.
7.	As a requester's manager, I approve updated purchase requisition.
8.	As a purchase agent, I have a possibility to return purchase requisition to requester, because of missing information.
9.	As a requester's manager, I have possibility to reject purchase requisition.
10.	Purchase requisition approval can be done manually or automatically.
11.	As a client, I request the system to be easy to use.

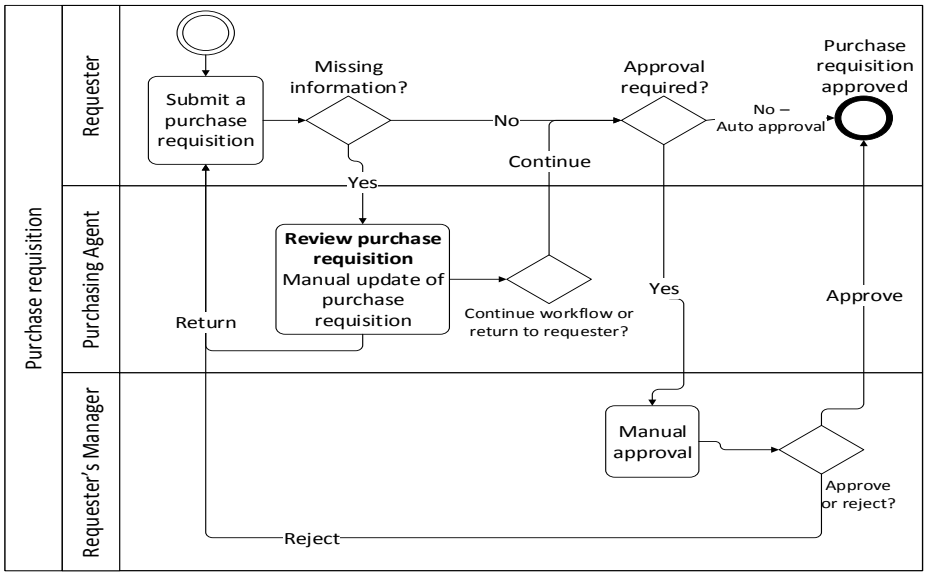


Fig 8. Activity diagram of purchase requisition review and approval management system

Activity diagram was adapted from (Dynamics365 Purchase requisition workflow) to visualize the workflow.

The next step was to identify Enterprise Metamodel elements by mapping Enterprise Metamodel elements with the activity diagram elements.

Table 9. Real case scenario – purchase review and approval management system

EM element \ Real case element		Requester	Purchasing Agent	Requester's Manager	Initial Node	Purchase Request Activation	Submitted purchase	Decision Node	Missing information Flow	No missing information	Purchase review Activation	Manual update of purchase	Decision Node	Continue Flow	Return to requester Flow	Decision Node	Approval required Flow	Auto-approval Flow	Approval Activation	Manual approval	Decision Node	Approval Flow	Rejection Flow	Activity Final Node
Actor		+	+	+																				
Event						+					+								+					
Process	Material Input						+																	
	Material Output											+								+				
Function	Business Rules				+			+					+			+					+			+
	Informa-tion Flow								+	+				+	+		+	+				+	+	
	Informat ion Activity					+						+								+				

The next step was to identify Requirements Ontology elements by capturing user stories into Requirements Ontology elements, identifying which sentence or expression belongs to which element of the ontology structure.

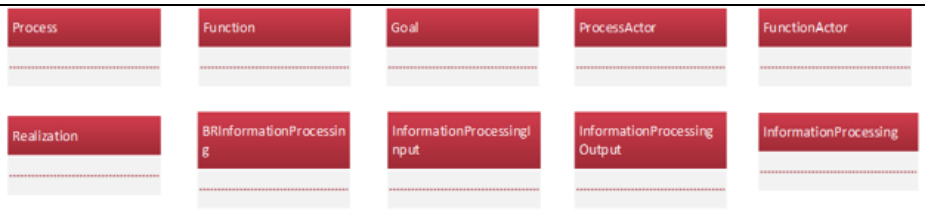
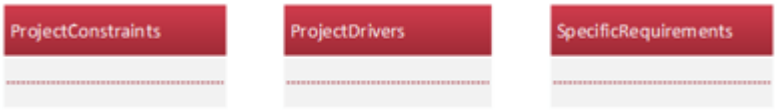
Table 10. Requirements Ontology elements - purchase requisition review and approval management system

Real case element	**Requirements Ontology element**
Purchase requisition review and approval management system.	Definition
Statements: New platform is created; Three steps needed – submit, review, approve; Interaction among 3 actors, 4 data elements; 7 task elements and 9 process elements.	Scope
Statements: • Requester must have a possibility to request a purchase requisition. • Purchasing agent must have a possibility to receive purchase requisition. • Purchasing agent must have possibility to review purchase requisition. • Purchasing agent must have a possibility to update purchase requisition. • Purchasing agent must have a possibility to approve purchase requisition automatically.	Functional requirements

• Purchasing agent must have a possibility to send for approval to requester's manager. • Requester's manager must have a possibility to approve updated purchase requisition. • Purchase agent must have a possibility to return purchase requisition to requester, because of missing information. • Requester's manager must have possibility to reject purchase requisition. • Purchase requisition approval can be done manually or automatically.	
Statements: • System must be easy to use. • System must be accessible for all three user via internet.	Non-functional requirements
• 3 actors: Requester; Purchasing Agent; Requester's Manager. • 4 data elements: Submitted purchase; Reviewed purchase; Approved purchase; Rejected purchase. • 7 task elements: Purchase requisition request activation; Purchase review activation; Manual update of purchase; Purchase approval activation; Manual approval; Auto approval; Purchase rejection. • 9 process elements: Initial Node; • 4 Decision Nodes: Missing information Flow; Non-missing information Flow; Continue Flow; Return to requester Flow; Approval required Flow; Approval Flow; Rejection Flow.	Data requirements

After all elements were identified, they were put in the EM and Requirements Ontology structural schema to identify which elements were covered and what is missing.

Table 11. Supplementing elements

The knowledge in the Enterprise Metamodel supplements requirements with these elements:

Requirements Ontology supplements requirements with these elements:


Two similar processes where put in our method structure to identify elements. They differ on the scope of the system, on the elements and the flow, but domain is similar.

One have integration elements, another have three steps needed and more decision nodes.

To conclude the basic case study, we can assume, that it showed that even upper-level ontology and enterprise model structures were used, not going deeper into quality rules applying, but the structure already helps System Analyst to identify the missing gaps in the requirements and help to overview and validate requirements to come closer to better quality. Enterprise Metamodel complements Requirements Ontology by supplementing it with domain knowledge elements. Also, it helps to formalize requirements generating process, shows how important it is to have as much information as possible gathered in a specific structure.

5. Conclusions and future works

Future works will include the research and experiments with deeper layers of the method. Requirements Ontology will be extended and it's components (classes, properties, instances, relations and etc.) will be described. It is planned, that Enterprise Metamodel will be verified and transformed into Enterprise Ontology for a better validation with Requirements Ontology and data reasoning. Also IEEE 830 standard requirements quality criteria rules will be applied to Knowledge repository for requirements validation, for the method to be complete.

Based on the MDA methodology and Ontology Definition Metamodel (ODM) 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. The method itself, lets the requirements expressed by the potential user to be summarized by the analyst by using the method. It gives the requirements engineering process support tool for specifying the requirements. It helps to specify requirements meeting the IEEE 830 standard criteria, by using domain knowledge and requirements relations in the ontology support.

The case study showed that ontology and enterprise modelling technologies can be used together for better requirements engineering results. This approach defines the structure to requirements, identifies elements and relationships between elements in requirements. Enterprise model gives deeper understanding about the domain, in our case – order management, identifies missing elements from the data we get in requirements elicitation, in our case – simple user stories. Enterprise Model also can help to generate activity diagrams for better understanding about the domain. This case study showed also that using ontologies complements the Enterprise Model technology as ontology identifies requirements itself, also the scope of the system creation project, Requirements Ontology specifically identifies elements needed for complete requirements specification.

The case study was performed on higher level of modelling, using UML elements, identifying upper level objects. The result was Requirements Ontology and Enterprise Metamodel mapping with incorporated real world case elements in the mapping and identifying existing objects and missing objects. Ontologies and enterprise modelling can be mapped together to get better results for requirements improval purposes, it supports System Analyst for designing software requirements specification.

References

Alonso, J. B. (2006). Ontology-based Software Engineering, Engineering Support for Autonomous Systems. ASLab-ICEA-R-2006-016 v 0.1 Draft of 2006-11-15.

Castañeda, V., Ballejos, L., Caliusco, M. L., Galli, M. R. (2010) The Use of Ontologies in Requirements Engineering. Global Journal of Researches in Engineering, 10:2–8.

Castro, J., Kolp, M., Mylopoulos, J. (2002) Towards Requirements-Driven Information Systems Engineering: The Tropos Project. Information Systems, 27(6).

Dardenne, A., Lamsweerde, A., Fickas, S. (1993) Goal-directed Requirements Acquisition. Sci. Comput. Program., 20:3–50.

Dynamics 365, Dynamics 365 for Finance and Operations, Microsoft. Purchase requisition workflow. Available online: *https://docs.microsoft.com/en-us/dynamics365/unified-operations/supply-chain/procurement/purchase-requisitions-workflow*

Fadel, G., Fox, M., Gruninger, M. (1994). A Generic Enterprise Resource Ontol-ogy. In: Proceedings of the 3rd Workshop on Enabling Technologies: Infrastruc-ture for Collaborative Enterprises. p. 117-128

Farfeleder, S., Moser, T., Krall, A., Stålhane, T., Omoronyia, I., Zojer, H. (2011). Ontology-driven Guidance for Requirements Elicitation. In Proceedings of the 8th Extended Semantic Web Conference on The Semantic Web: Research and Applications - Volume Part II, ESWC'11, pages 212–226, Berlin, Heidelberg, 2011. Springer-Verlag.

Firesmith, D. (2003) Specifying Good Requirements. Journal of Object Technology, 2:77–87.

Ghaisas, S., Ajmeri, N. (2012) Knowledge-assisted Ontology-based Requirements Evolution, Tata Research Development and Design Centre (TRDDC), Pune, India

Gudas, S. (2009) 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).ISBN -13: 978-1-9934272-61-9 (Volume III)

Haruhiko, K., Motoshi, S. (2005) Ontology Based Requirements Analysis: Lightweight Semantic Processing Approach. In Proc. Fifth International Conference on Quality Software (QSIC 2005), 2005.

IBM software requirements template. Available online: *https://www.ibm.com/developerworks/.../files/.../document/.../SRS_Sample.doc*

IEEE 830 – 1998 standard. IEEE Recommended Practice for Software Requirements Specifications. Available online: *http://www.cse.msu.edu/~cse870/IEEEXplore-SRS-template.pdf*

ISO/IEC 25001:2014. Available online: *https://www.iso.org/standard/64787.html*

Jureta, I. J., Mylopoulos, J., Faulkner, S. (2009) A Core Ontology for Requirements. Appl. Ontol., 4(3-4):169–244, 2009.

Khan, N. A. (2011) Transformation of Enterprise Model to Enterprise Ontology. Tekniska Hogskolan, Jonkoping, Sweden.

Kof, K. (2004) Natural Language Processing For Requirements Engineering Applicability. Available online: *https://pdfs.semanticscholar.org/887f/1e32845851b70e432c500164da7741637219.pdf*

Kossmann, M., Wong, R., Odeh, M., Gillies, A. (2008) 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.

Levendovszky, T., Karsai, G., Maroti, M., Ledeczi, A., Charaf A. (2002) Model reuse with meta model-based transformation,Springer-Verlag Berlin Heidelberg

Lopata, A. (2004) Disertacija. Veiklos modeliu grindžiamas kompiuterizuotas funkcinių vartotojo reikalavimų specifikavimo metodas. Summary available online: *http://talpykla.elaba.lt/elaba-fedora/objects/elaba:2012661/datastreams/MAIN/content*

Lopata, A., Ambraziūnas, M., Gudas, S., Butleris, R. (2012) „The Main Principles of Knowledge-Based Information Systems Engineering“, Electronics and Electrical Engineering, Vol. 11, No 1 (25), pp. 99-102, ISSN 2029-5731.

Lopata, A., Ambraziūnas, M., Gudas, S. (2012) Knowledge-based MDA requirements specification and validation technique. Transformations in Business & Economics, 2012, 11(1(25)), 248-260. ISSN 1648-4460.

Lopata, A., Veitaitė, I., Gudas, S., Butleris, R. (2014) Case tool component – subsystem, UML diagrams generation process, Transformations in Business & Economics.2014, Vol. 13 Issue 2B, p676-696. ISSN1648-4460

Makrickienė, N., Lopata, A. (2018) Requirements Engineering, Supported by Ontology and Enterprise Modelling, ICYRIME 2018, ISSN1613-0073, Vol-2152

Morkevicius, A., Gudas, S. (2011) "Enterprise Knowledge Based Software Re-quirements Elicitation", Information Technology and Control, Vol. 40, No 3, pp. 181-190, 1392 – 124X

OMG MOF. (2015). OMG Meta Object Facility (MOF) Core Specification. Version 2.4.2. April 2014, Available online: *http://www.omg.org/spec/MOF/2.4.2*

Riechert, T., Lauenroth, K., Lehmann, J. (2007) Semantic Requirements Engineering. In Proceedings of the SABRE-07 SoftWiki Workshop.

Sancho, P. P., Juiz, C., Puigjaner, R., Chung, L., Subramanian, N. (2007) An Approach to Ontology-aided Performance Engineering Through NFR Framework. In WOSP '07: Proceedings of the 6th international workshop on Software and performance, pages 125–128, New York, NY, USA, 2007. ACM Press.

Siegemund, K., Thomas, E. J., Zhao, Y., Assmann, U. (2010) Towards Ontology-driven Requirements Engineering. Available online: *http://iswc2011.semanticweb.org/fileadmin/iswc/Papers/Workshops/SWESE/4.pdf*

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

Xuefeng, Z. (2005) Inconsistency Measurement of Software Requirements Specifications: An Ontology-Based Approach. In ICECCS '05: Proceedings of the 10th IEEE International Conference on Engineering of Complex Computer Systems, pages 402–410, Washington, DC, USA, 2005. IEEE Computer Society.

Ying-ying, Y., Zong-yon, L., Zhi-xue, W. (2008) Domain Knowledge Consistency Checking for Ontology-Based Requirement Engineering. In CSSE '08: Proceedings of the 2008 International Conference on Computer Science and Software Engineering, pages 302–305, Washington, DC, USA, 2008. IEEE Computer Society.

W3C World Wibe Web Consortium, OWL Working Group. OWL 2 Web Ontology Language Document Overview, Release date: 2012/12/01. Available online: *http://www.w3.org/TR/2012/REC-owl2-overview-20121211/*

W3C World Wibe Web Consortium, Ontology Driven Architectures and Potential Uses of the Semantic Web in Systems and Software Engineering, Release date: 2006/02/11. Available online: *https://www.w3.org/2001/sw/BestPractices/SE/ODA/*

Received February 10, 2019, revised April 4, 2019, accepted April 4, 2019