Knowledge-Based UML Use Case and UML Activity Models Generation from Enterprise Model. School of Languages Case Study

Ilona Veitaitė[a] and Audrius Lopata[b]

[a] *Vilnius University, Kaunas Faculty, Muitinės str. 8, Kaunas, LT-44280, Lithuania*
[b] *Kaunas University of Technology, Studentų str. 50, Kaunas, LT-51368, Lithuania*

Abstract

The main purpose of this paper is to present knowledge-based Enterprise model (EM) advantages as data repository for UML models generation. UML models can be generated from Enterprise Model, whose main requirement is that gathered data in this model would be verified and validated by analyst. UML models generation process is implemented by using particular transformation algorithms presented in previous researches. In this paper generation process from EM is represented by the School of Languages case study. By using problem domain data UML Use Case and UML Activity models are generated. Generation results help to define advantages of Enterprise Model usage as storage for problem domain data.

Keywords 1

Enterprise Model, UML, IS Engineering, Activity, Use Case, Knowledge-based.

1. Introduction

Today's IT professionals such as analysts, designers, developers still face with challenges of IS engineering process. First phases of IS development life cycle is also quite difficult, especially, enterprise modelling, when all gathered data is specified and prepared for IS design phase. According the efforts put in this phase success of final IS depends. There are many various types, standards of models used in design phase and very great impact for IS creation success has Enterprise model chosen for this process [1][2]. UML is one of mostly used standard for IS design among professionals of IS engineering field. According to UML models prepared in design phase IS code may be generated, different design patterns applied, impact and complexity analysis accomplished [1][6][8]. All advantages of UML models usage may be fulfilled only if data used for their design is verified and validated. This fulfillment can be ensured by Enterprise model. Enterprise model - knowledge-based repository, where problem domain data of enough quality is stored. By using transformation algorithms all UML models can be generated from Enterprise Model [3][5][7][9][10]. Structure of particular Enterprise Meta-Model (EMM) and Enterprise Model as the background for the research in this paper is presented almost two decades ago. Previous researches in this field are designated to prove that Enterprise meta-model and Enterprise Model is enough for different types of models generation in IS design phase [10][11][12][13][14].

2. Description of Knowledge-Based Enterprise Model

EMM is formally defined Enterprise Model structure, which consists of a formalized EM in line with the general principles of control theory. EM is the main source of the necessary knowledge of the particular business domain for IS engineering and IS re-engineering processes (Figure 1) [3][4].

IVUS2021: Information Society and University Studies 2021, April 23, 2021, Kaunas, Lithuania
EMAIL: ilona.veitaite@knf.vu.lt (A. 1); audrius.lopata@ktu.lt (A. 2)

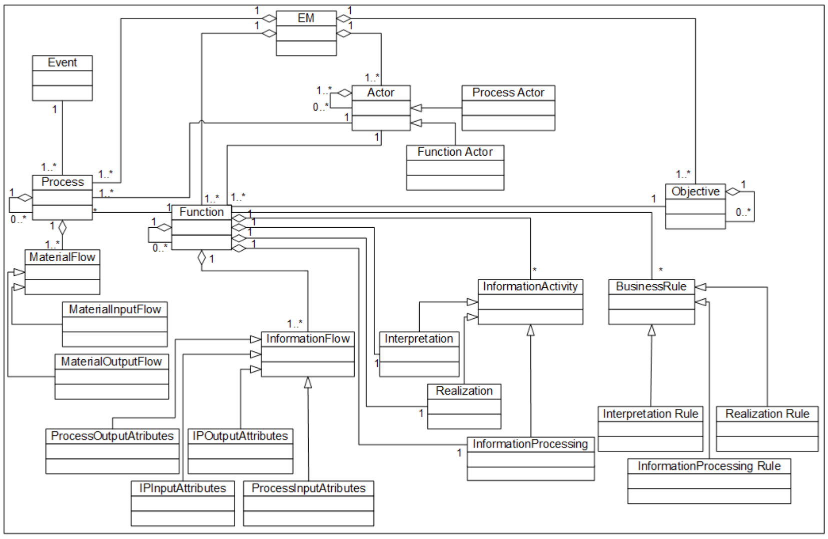

Figure 1: Enterprise Meta-Model class diagram [3][4][10]

EM class model has twenty-three classes. Essential classes are Process, Function and Actor. Class Process, Function, Actor and Objective can have an internal hierarchical structure. These relationships is presented as aggregation relationship. Class Process is linked with the class MaterialFlow as aggregation relationship. Class MaterialFlow is linked with the classes MaterialInputFlow and MaterialOutputFlow as generalization relationship. Class Process is linked with Classes Function, Actor and Event as association relationship. Class Function is linked with classes InformationFlow, InformationActivity, Interpretation, InformationProcessing and Realization as aggregation relationship. These relationships define the internal composition of the Class Function. Class InformationFlow is linked with ProcessOutputAtributes, ProcessInputAtributes, IPInputAttributes and IPOutputAttributs as generalization relationship. Class InformationActivity is linked with Interpretation, InformationProcessing and Realization as generalization relationship. Class Function linked with classes Actor, Objective and BusinessRule as association relationship. Class BusinessRule is linked with Interpretation Rule, Realization Rule, InformationProcessing Rule as generalization relationship. Class Actor is linked with Function Actor and Process Actor as generalization relationship [3][4][5][11].

3. Transformation Algorithms from Knowledge-Based Enterprise Model

Each of static and dynamic UML models can be generated through transformation algorithm and each of UML models has separate transformation algorithm. These transformation algorithms are defined in previous researches. Main focus of researches is designated for generation behavioural or dynamic UML models, because their complexity and variability [10][11][12][13][14][15].

3.1. UML Use Case Model Transformation Algorithm

UML Use Case model generation from Enterprise Model transformation algorithm starts from initial element. In this generation process Actor/Subject is initial element, after actor/subject element is generated, Use Case element is selected and generates, then Include, Extend and Association relationships elements are selected and generated. Transformation algorithm is illustrated by following steps [12]:

- Step 1: The initial element Actor from Enterprise Model for UML Use Case model generation is selected.
- Step 2: If Actor element is initial element of UML Use Case model, then Actor element is generated, else Subject element is generated.

- Step 3: Process element from Enterprise model, which is related with the initial Actor element is selected.
- Step 4: If Process element is Use Case element related to Actor/Subject, then Use Case element is generated, else Function element is selected.
- Step 5: Function element is generated as Use Case element.
- Step 6: Business Rule element as link of Actor/Subject element from Enterprise Model which is related with the Process/Function element is selected.
- Step 7: If Business Rule element is UML Use Case model's simple Association element and serves as link between Actor/Subject and Process/Function elements then Association is generated from Enterprise model, else if it is Extend element, then Extend element is generated from Enterprise model, else Include element is generated from Enterprise model.
- Step 8: There is checking if there are more Business Rules in Enterprise Model related to UML Use Case model. In case, there are, algorithm goes back to step 3.
- Step 9: UML Use Case elements Actor/Subject and Process/Function are linked according to Business Rules.
- Step 10: UML Information flow element Actor/Subject is updated.
- Step 11: There is checking if there are more Actors/Subject elements in Enterprise Model related to UML Use Case model. In case, there are, algorithm goes back to step 1.
- Step 12: Else all UML Use Case model elements and links are generated from Enterprise Model.

3.2. UML Activity Model Transformation Algorithm

UML Activity model describes how activities are coordinated to provide a service. UML Activity model from EM transformation algorithm is described by following steps [14]:

- Step 1: Particular UML model for generation from EM process is identified and selected.
- Step 2: If the particular UML model for generation from EM process is selected then algorithm process is continued, else the particular UML model for generation from EM process must be selected.
- Step 3: First element from EM is selected for UML model, identified previously, generation process.
- Step 4: If the selected EM element is initial UML model element, then initial element is generated, else the other EM element must be selected (the selected element must be initial element).
- Step 5: The element related to the initial element is selected from Enterprise model.
- Step 6: The element related to the initial element is generated as UML model element.
- Step 7: The element related to the previous element is selected from Enterprise model.
- Step 8: The element related to the previous element is generated as UML model element.
- Step 9: If there are more related elements, then they are selected from EM and generated as UML model elements one by one, else the link element is selected from Enterprise model.
- Step 10: The link element is generated as UML model element.
- Step 11: If there are more links, then they are selected from EM and generated as UML model elements one by one, else the Business Rule element is selected from Enterprise model.
- Step 12: The Business Rule element is generated as UML model element.
- Step 13: If there are more Business Rules, then they are selected from EM and generated as UML model elements one by one, else the generated UML model is updated with all elements, links and constraints.
- Step 14: Generation process is finished.

4. School of Languages Case Study

The School of languages management process may be defined as courses management system that is used to manage language courses, timetable, lecturers and participants of the courses.

The School of languages offers a list of language courses related with different level of language knowledge and diverse study methods, which improve speaking, reading, listening, speaking and grammar skills. Each language course is made up of set of topics related with skills dedicated for improvement. The School of languages publish and maintain a timetable of different language courses and appoints lecturers every year.

Lecturers in the school of languages are appointed courses to teach according to the language knowledge level and their availability. There is a course administrators in the school of languages to manage the courses including course content, courses appointments to lecturers and preparation of the course timetable. Participants of language courses may review suggested courses, get information about lecturers and check the timetable of the courses.

The School of Languages purpose is to use the courses management information system to improve their services by managing courses, timetable, lecturers and participants.

4.1. School of Languages UML Use Case Model

The main concept of UML Use Case model is that it assists to design a system from the end user's perspective, in this case, end users are Administrator, Lecturer and Participant.

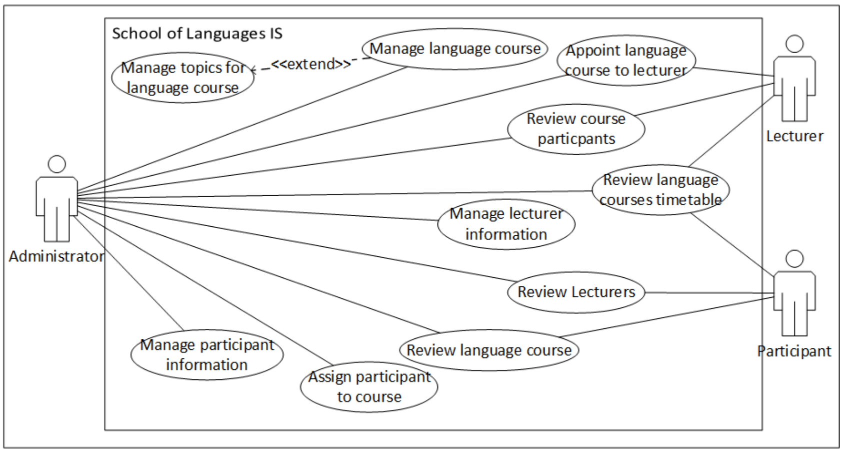
School of Languages IS
Manage topics for language course
<<extend>>
Manage language course
Appoint language course to lecturer
Review course particpants
Review language courses timetable
Manage lecturer information
Review Lecturers
Manage participant information
Review language course
Assign participant to course
Administrator
Lecturer
Participant

Figure 2: UML Use Case Model.

Figure 2 presents generated UML Use Case model from Enterprise model, where uses cases of each actor are displayed. Descriptions of these generated UML Use Case Model elements are presented in the Table 1.

Table 1
Generated UML Use Case Model elements from EM with descriptions [10] [12]

EM elements	UML Use Case Model elements	Description
Actor	Actor: Administrator	Administrator has full access to all use cases implemented in School of languages IS.
	Actor: Lecturer	Lecturer can check appointed to him language course, review list of participants of the appointed

		language course and review his timetable.
	Actor: Participant	Participant is a student of chosen language courses, he can review list of courses, review information about lecturers and also review timetable of chosen courses.
Business Rules	Association	Relationship that links all actors to the use cases they may access
	Extend	Relationship that links use case, which may be done together with another use case. During the course management process, topics may be added
Process / Function	Use Case: Manage language course Use Case: Manage topics for language course Use Case: Appoint language course to lecturer Use Case: Review course participants Use Case: Review language courses timetable Use Case: Manage lecturer information Use Case: Review lecturers Use Case: Review language course Use Case: Assign participant to course Use Case: Manage participant information	Use cases which are accessible to the actors.

4.2. School of Languages UML Activity Models

UML Activity models present coordination of activities regarding process participants: Administrator, Lecturer and Participant. There can be more UML Activity model generated and at different levels of abstraction, but in this paper three possible models are presented, only to present this possibility.

4.2.1. UML Activity Model. Actor: Administrator

UML Activity model for language course management performed by Administrator is generated from Enterprise model.

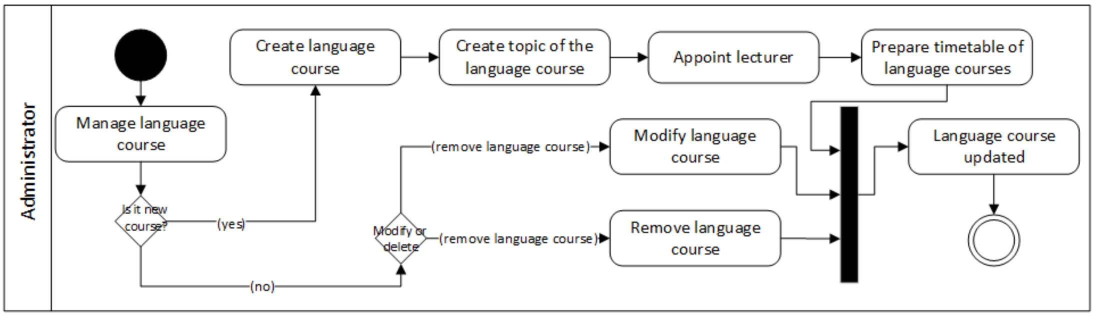

Figure 3: UML Activity Model. Actors: Administrator

Figure 3 presents generated UML Activity model from Enterprise model, where activities of actor, in this case, Administrator are displayed. Descriptions of these generated UML Activity Model elements are presented in the Table 2.

Table 2
Generated UML Activity Model elements from EM with descriptions [10] [14]

EM elements	UML Activity Model elements	Description
Actors	Swimlane: Administrator	Administrator starts the process and performs all presented activities.
Process / Function	Activity: Manage language course Activity: Create language course Activity: Create topic of the language course Activity: Appoint lecturer Activity: Prepare timetable of language course Activity: Modify language course Activity: Remove language course Activity: Language course updated	All activities performed by Administrator of the particular process.
Information Flow	Object Flows	All flows between activities, show sequence of activities flow.
Business Rules	Control Node: Join Node	Joins three different activities related with language course management.
	Control node: Initial Node	Starts the process.
	Control node: Decision Node	Two decision nodes: checks course status; another allows modify or remove the course.
	Control node: Final Node	Finishes process.

4.2.2. UML Activity Model. Actors: Administrator and Lecturer

UML Activity model for lecturer information management performed by Administrator and Lecturer is generated from Enterprise model.

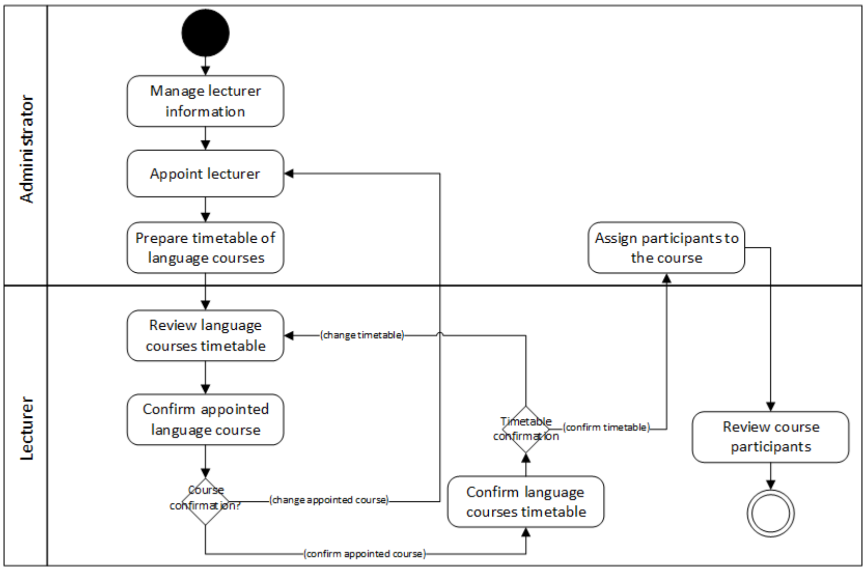
Administrator
Lecturer
Manage lecturer information
Appoint lecturer
Prepare timetable of language courses
Assign participants to the course
Review language courses timetable
(change timetable)
Confirm appointed language course
Timetable confirmation
(confirm timetable)
Review course participants
Course confirmation?
(change appointed course)
Confirm language courses timetable
(confirm appointed course)

Figure 4: UML Activity Model. Actors: Administrator and Lecturer

Figure 4 presents generated UML Activity model from Enterprise model, where activities of actors, in this case, Administrator and Lecturer are displayed. Descriptions of these generated UML Activity Model elements are presented in the Table 3.

Table 3
Generated UML Activity Model elements from EM with descriptions [10] [14]

EM elements	UML Activity Model elements	Description
Actors	Swimlane: Administrator	Administrator starts the process for lecturer information management; appoints lecturer to the language course, prepares timetable and assigns participants to the language course.
	Swimlane: Lecturer	Lecturer may review language courses timetable, confirms appointed language courses, confirms timetable and may review course participants.
Process / Function	Activity: Manage lecturer information Activity: Appoint lecturer Activity: Prepare timetable of language courses Activity: Review language course timetable Activity: Confirm appointed language course Activity: Confirm language courses timetable Activity: Assign participants to the course Activity: Review course participants	All activities performed by Actors of the particular process.
Information Flow	Object Flows	All flows between activities, show sequence of activities flow.
Business Rules	Control node: Initial Node	Starts the process.
	Control node: Decision Nodes	Two decision nodes: one confirms appointed course; another confirms course timetable.
	Control node: Final Node	Finishes process.

4.2.3. UML Activity Model. Actors: Administrator and Participant

UML Activity model for participant information management performed by Administrator and Participant is generated from Enterprise model.

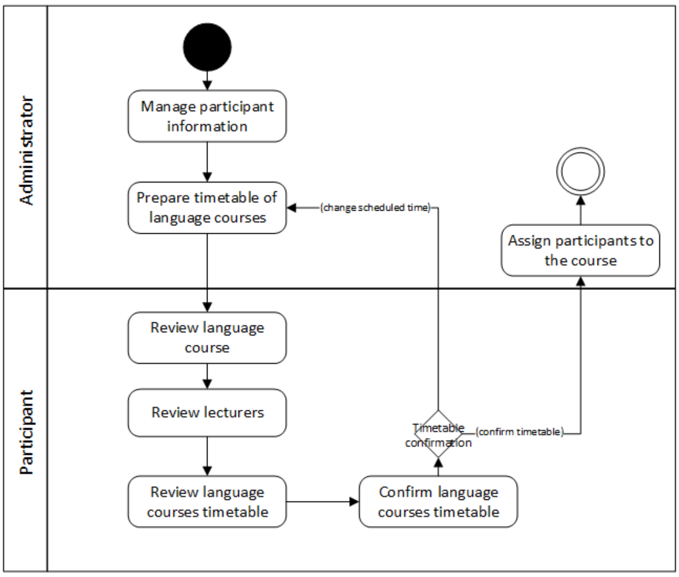
Administrator
Participant
Manage participant information
Prepare timetable of language courses
(change scheduled time)
Assign participants to the course
Review language course
Review lecturers
Timetable confirmation
(confirm timetable)
Review language courses timetable
Confirm language courses timetable

Figure 5: UML Activity Model. Actors: Administrator and Participant

Figure 5 presents generated UML Activity model from Enterprise model, where activities of actors, in this case, Administrator and Participant are displayed. Descriptions of these generated UML Activity Model elements are presented in the Table 4.

Table 4

Generated UML Activity Model elements from EM with descriptions [10] [14]

EM elements	UML Activity Model elements	Description
Actors	Swimlane: Administrator	Administrator starts the process of participant information management; prepares timetable of the courses and assigns participants to the course.
	Swimlane: Participant	Participant may review the courses descriptions, review lecturers responsible for the course, review timetable and confirm it.
Process / Function	Activity: Manage participant information Activity: Prepare timetable of language courses Activity: Review language course Activity: Review lecturers Activity: Review language courses timetable Activity: Confirm language courses timetable Activity: Assign participants to the course	All activities performed by Actors of the particular process.
Information Flow	Object Flows	All flows between activities, show sequence of activities flow.
Business Rules	Control node: Initial Node	Starts the process.
	Control node: Decision Node	One decision node: confirms course timetable
	Control node: Final Node	Finishes process.

5. Results

Main activities of the analyst responsibility may be described as: analysis of enterprise situation, identification of opportunities for improvements, design information system which will add value to the enterprise. The analyst gathers problem domain information, identifies all requirements, searches for suitable meta-model, verifies and validates data and starts implement IS by designing UML or other models. There is always risk of new problem, information or requirements appearance, which cause difficult design models updating and improving process. In this case, duration of IS developing process increase, and number of errors grows. Using Enterprise Model as the prime problem domain knowledge repository in IS engineering process ensures correctness and quality of generated IS design models after any possible problem domain data update (Table 5).

Table 5
Comparison of IS Analyst's activities by criteria

Criteria	IS Analyst	Enterprise Knowledge-Based UML Dynamic Models Generation Method
Gathering problem domain data	Yes	Yes
Requirements identification	Yes	Yes
Data preparation for modelling	Yes	Yes
Project models design	Yes	No
Problem domain data update	Yes	Yes
Project model design improvement	Yes	No
Increased IS development process duration	Yes	No
Increased number of errors	Yes	No
Increased number of errors	Yes	No

By using Enterprise Model in IS engineering process the analysts enters all gathered problem domain data into EM. Problem domain knowledge stored in EM is used for UML model generation through transformation algorithms. After any possible new data upload to the EM, it is re-used and new UML models based on improved data are generated. There is no need for the analyst to re-do entire process of models design.

6. Conclusions

The first part of the paper presents the Enterprise model structure and description of UML Use Case and UML Activity models transformation algorithms for generation from EM depicted by steps.

The next part presents School of Languages case study, which data is stored in knowledge-based Enterprise Model and is used in two types UML models generation process. There are presented UML Use Case model and three UML Activity models of different Actors perspective generated from Enterprise Model. These generated models confirms, that data stored in Enterprise model is enough for generation process.

Final part describes comparison by certain criteria of information systems analyst's activities during IS development process with and without usage of Enterprise knowledge-based UML models generation method.

7. References

[1] Dunkel J., Bruns R. (2007) Model-Driven Architecture for Mobile Applications. Proceedings of the 10th Inter-national Conference on Business Information Systems (BIS), Vol. 4439/2007, pp. 464–477.

[2] Eichelberger H., Eldogan, Y., Schmid K. A (2011) Comprehensive Analysis of UML Tools, their Capabilities and Compliance. Software Systems Engineering. Universität Hildesheim. versio 2.0

[3] 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, Orlando, Florida, USA, Vol. III, p.161-266 ISBN -10: 1-9934272-61-2 (Volume III).ISBN -13: 978-1-9934272-61-9

[4] Gudas S., (2012) Informacijos sistemų inžinerijos teorijos pagrindai/ Fundamentals of Information Systems Engineering Theory. (Lithuanian)Vilnius University. ISBN 978-609-459-075-7

[5] Jacobson, I., Rumbaugh, J., Booch, G. (2005) Unified Modeling Language User Guide, The Second Edition. Addison-Wesley Professional. ISBN: 0321267974

[6] Jenney J. (2010) Modern Methods of Systems Engineering: With an Introduction to Pattern and Model Based Methods. ISBN-13:978-1463777357

[7] OMG UML (2021) Unified Modeling Language version 2.5.1. Unified Modelling https://www.omg.org/spec/UML/About-UML/ .

[8] Sajja, P.S., Akerkar, R. (2010) Knowledge-Based Systems for Development. Advanced Knowledge Based Systems: Model, Applications & Research, Vol. 1

[9] UML Diagrams (2021) UML diagrams characteristic. www.uml-diagrams.org

[10] Veitaitė, I., Lopata, A. (2017) Transformation algorithms of knowledge based UML dynamic models generation // Business information systems workshops BIS 2017, Poznan, Poland, 28-30 June / editor Witold Abramowicz . - Series : Lecture notes in business information processing. Vol 303. Cham : Springer International Publishing, 2017

[11] Veitaitė, I., Lopata, A. (2018) Problem domain knowledge driven generation of UML models // Information and software technologies: 24th International Conference, ICIST 2018, Vilnius, Lithuania, October 4–6, 2018: proceedings / edited by: Robertas Damaševičius, Giedrė Vasiljevienė. Cham : Springer.

[12] Veitaitė, Ilona; Lopata, Audrius. Knowledge-based UML use case model transformation algorithm // Business information systems workshops: BIS 2019 international workshops, Seville, Spain, June 26–28, 2019: revised papers / W. Abramowicz, R. Corchuelo (eds.). Cham : Springer, 2019. ISBN 9783030366902. eISBN 9783030366919. p. 39-48. (Lecture notes in business information processing, ISSN 1865-1348, eISSN 1865-1356 ; Vol. 373). DOI: 10.1007/978-3-030-36691-9.

[13] Veitaite I., Lopata A. (2020) Knowledge-Based Transformation Algorithms of UML Dynamic Models Generation from Enterprise Model. In: Dzemyda G., Bernatavičienė J., Kacprzyk J. (eds) Data Science: New Issues, Challenges and Applications. Studies in Computational Intelligence, vol 869. Springer, Cham

[14] Veitaitė, Ilona; Lopata, Audrius. Knowledge-based UML activity model transformation algorithm // Information society and university studies 2020 : proceedings of the information society and university studies, 2020, Kaunas, Lithuania, April 23, 2020 / edited by: Adrius Lopata, Vilma Sukackė, Tomas Krilavičius, Ilona Veitaitė, Marcin Woźniak. Aachen : CEUR Workshop Proceedings. ISSN 1613-0073. 2020, p. 114-120. (CEUR Workshop Proceedings, ISSN 1613-0073 ; vol. 2698).

[15] Veitaite I., Lopata A. (2020) Knowledge-Based Generation of the UML Dynamic Models from the Enterprise Model Illustrated by the Ticket Buying Process Example. In: Lopata A., Butkienė R., Gudonienė D., Sukackė V. (eds) Information and Software Technologies. ICIST 2020. Communications in Computer and Information Science, vol 1283. Springer, Cham. https://doi.org/10.1007/978-3-030-59506-7_3