Knowledge Based UML Information Flow Model Transformation Algorithm

Ilona Veitaite
Institute of Applied Informatics
Vilnius University, Kaunas
Faculty
Kaunas, Lithuania
Ilona.Veitaite@knf.vu.lt

Audrius Lopata
Institute of Applied Informatics
Vilnius University, Kaunas
Faculty
Kaunas, Lithuania
Audrius.Lopata@knf.vu.lt

***Abstract* – The main scope is to present UML Information Flow model generation from Enterprise model (EM) transformation algorithm. The transformation algorithm description is presented in details and depicted steps. Whole generation process steps are illustrated by particular example following the transformation algorithm step by step.**

***Keywords* –** *Enterprise Model, Knowledge-based, IS Engineering, UML, Information Flow, Information Item.*

I. INTRODUCTION

There have been quite many attempts for the analysis of UML models generation from different knowledge based models combining other modelling languages, workflow patterns and frameworks or even generation from natural language specifications [1, 2, 8].

UML models are receiving an increasing attention from researchers in the recent years. It is a very challenging target for analysis of UML models since the knowledge about an enterprise system is allocated within several model views. UML models are maintained to decrease the confusion of the problem with the increase enterprise changes. By operating UML models knowledge can be effectively expressed and can be used simply in all phases of IS development life cycle [2, 6, 9, 18].

UML as one of the main components of IS development life cycle phase models, can be generated in semi-automatic way from knowledge repository – Enterprise model. This kind of realization will improve the efficiency of these participants of information system development process: system analyst and/or system designer and/or system developer.

II. ENTERPRISE MODEL ELEMENTS ROLE VARIATIONS

Enterprise meta-model is formally determined enterprise model composition, which contained of a formalized enterprise model alongside with the general principles of control theory. Enterprise model is the main source of the requisite knowledge of the specific problem domain for IS engineering and IS reengineering processes [3, 4, 5, 14, 23].

Enterprise meta-model manages Enterprise model composition. Enterprise model stores knowledge that is necessary for IS development process only and will be used during all phases of IS development life cycle [7, 14, 15, 16].

There is given formalized Enterprise meta-model description, which is needed to define UML Information Flow model generation process algorithm. Enterprise model can be described as Malcev algebra based algebra system (Fig. 1) [10, 19]:

$$M1=<K, R> \quad (1)$$

where M1 – Enterprise model as algebra system; K – elements set of M1 system; K={K1, K2,…, K21}, where K1,....K21 EM meta-classes; R – set of relationships between elements, where R={r1, r2, r3}.

For each set of K element Kn composition is defined as: Kn=<{an1, an2,…,ank}, {mn1, mn2,…,mnl}>, where {an1, an2,…,ank} – attributes of Kn element, {mn1, mn2,…,mnl}– methods of Kn element.

Enterprise model M1 composition is as follows:

$$M1=<\{K1, K2,...,K21\}, \{r1, r2, r3\}> \quad (2)$$

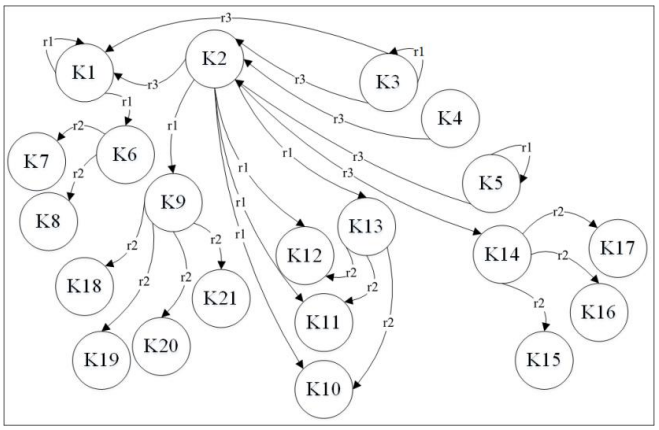


Fig. 1. *An Enterprise meta-model graphical schema based on Malcev algebra [10,19]*

where: K1 – meta-class Process, K2 – meta-class Function, K3 – meta-class Actor, K4 – meta-class Event, K5 – meta-class Goal, K6 – meta-class Material Flows, K7 – meta-class Input Material Flow, K8 – meta-class Output Material Flow, K9 – meta-class Information Flow, K10 – meta-class Interpretation, K11 – meta-class Data Processing and Solution Making, K12 –

meta-class Realization, K13 – meta-class Information Activity, K14 – meta-class Business Rules, K15 – meta-class Interpretation Business Rules, K16 – meta-class Data Processing and Solution Making Business Rules, K17 – meta-class Realization Business Rules, K18 – meta-class Process Output, K19 – meta-class Information processing Input Attributes, K20 – meta-class Information processing Output Attributes, K21 – meta-class Process Input, r1 – Aggregation, r2 – Generalization, r3 – Association.

Information systems design methods indicates the disposition of systems engineering actions, i.e. how, in what order and what UML model to use in the IS development process and how to implement the process (Table 1). Majority of them are based on different types of models describing varying aspects of the system qualities. Meaning of each model can be defined individually, but more important is the fact that each model is the projection of the system. An inexperienced specialist can use UML models inappropriately and the description of the system will possibly be insufficient or even mistaken [11, 12, 13, 20].

TABLE I. ENTERPRISE MODEL BUSINESS RULES ELEMENTS ROLE VARIATIONS IN PART OF UML DYNAMIC MODELS

EM	UML Model element	UML Dynamic Model
Busines Rule	Extend	Use Case Model
	Include	Use Case Model
	Association	Use Case Model
	Control Nodes	Activity Model
	Time Constraint	Timing Model
	Destruction Occurrence	Timing Model
	…	…
	Pseudostate	State Machine Model
	…	…

Determining specific UML model and selecting the initial model element is reasonably meaningful, because further generating process relies on it. Many UML model elements iterates in different UML model, but these elements describe different aspects of the system. In example Enterprise model element Business rule has different signification in different UML models [11, 12, 13, 20].

III. TRANSFORMATION ALGORITHMS

When Enterprise model as enterprise knowledge storage allows – the stored knowledge is sufficient, validated and verified – to generate UML models applying transformation algorithms. This kind storage can be used not only for knowledge of the enterprise gathering, but also as a tool that minimizes IS reengineering volume of work if any changes occur in an enterprise. UML models generation from Enterprise model is implementation of knowledge based IS development life cycle design phase.

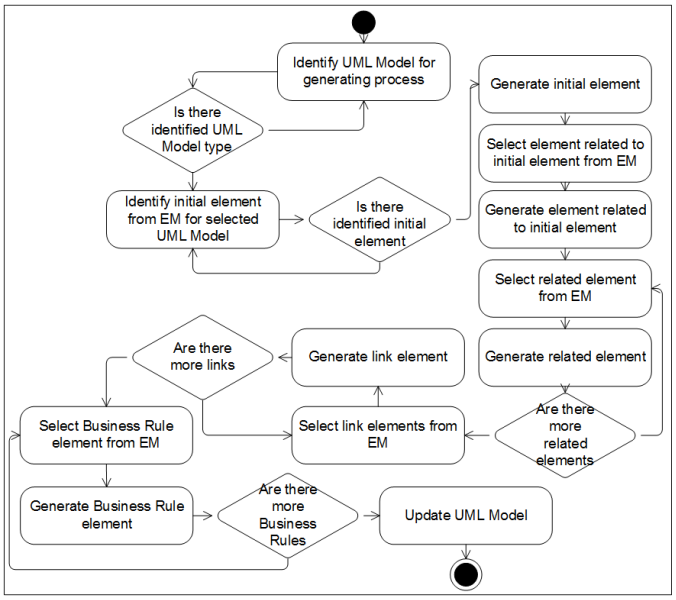


Fig. 2. *The top level transformation algorithm of UML models generation from EM process*

Transformation algorithm of UML models generation from Enterprise model is top level algorithm for enterprise meta-model based UML model generating process (Fig. 2). Main steps for generating process are identifying and selecting UML model for generating process, identifying starting (initial) element for the selected UML model and selecting all elements related to this UML model, generating enterprise model elements to UML model elements and generating the whole UML model.

Table 2 presents UML Information Flow model elements and its descriptions.

TABLE II. UML INFORMATION FLOW MODEL ELEMENTS DESCRIPTIONS [17, 21]

UML Information Flow Model element	Description
Actor	A dynamic classifier which specifies a role played by an external entity that interacts with the subject (e.g., by exchanging signals and data), a user of the designed system, some other system or hardware using services of the subject.
Class	A classifier which describes a set of objects that share the same: features, constraints, semantics (meaning).
Information flow	A directed relationship that is used as a specification of some kind of "information channel" for unidirectional transmission of information from sources to targets.
Information item	A classifier which represents some information transferred within a system from source(s) to target(s) of information flow and provides no details about the information they transfer as they do not have features.

There is given formalized UML Information Flow model description. UML Information Flow also can be described as Malcev algebra based algebra system (Fig. 3) [10,19]:

$$M4=<K,R> \quad (3)$$

where M4 – UML Information Flow model as algebra system; K – elements set of M4 system; K={K34, K35,…, K37}, where K34,....K37 UML Information Flow meta-classes; R – set of relationships between elements, where R={r1, r2, r3}.

UML Information Flow M4 composition is as follows:

$$M4=\{K34, K35,\ldots, K37\},\{r2\},\{r3\}> \quad (4)$$

where: K34 – meta-class Actor, K35 – meta-class Class, K36 – meta-class Information Flow, K37 – meta-class Information Item, r2 – Generalization, r3 – Association.

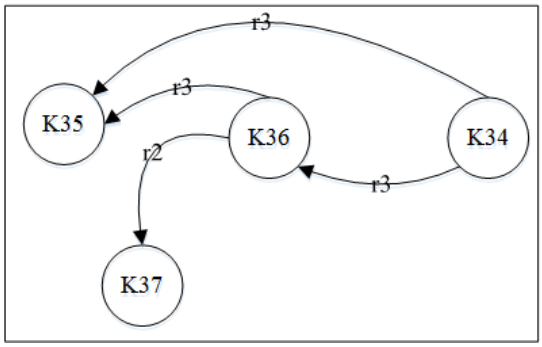


Fig. 3. *UML Information Flow graphical schema based on Malcev algebra*

According to the figure 3 it is clear that Enterprise model elements: Actor, Process, Function, Information Flow, Information processing Input Attributes, Information processing Output Attributes can be generated as UML Information Flow model elements: Actor, Class, Information Flow, Information Item.

TABLE III. INTERSECTION BETWEEN ENTERPRISE MODEL AND UML INFORMATION FLOW MODEL ELEMENTS

Enterprise model set element	UML Information Flow model set element	Formal description
Actor (K3)	Actor (K34)	φ1: K3→K34
Process (K1)	Class (K35)	φ1: K1→K35
Information processing Input Attributes (K19)	Information Item (K37)	φ2: K19→K37
Information processing Output Attributes (K20)	Information Item (K37)	φ2: K20→K37
Information Flow (K9)	Information Flow (K36)	φ6: K9→K36

Table 3 presents intersection between Enterprise model and UML Information Flow model elements, where formal description of Enterprise model elements generated to UML Information Flow model elements according to Malcev algebra can be found.

A. UML Information Flow Model Transformation Algorithm

UML Information Flow Model belongs to dynamic UML models part and shows exchange of information among system entities at some high levels of abstraction and it is directly related with UML Class and Use Case models (Fig. 4). This model describes information flows and provides information to Class and Use Case models.

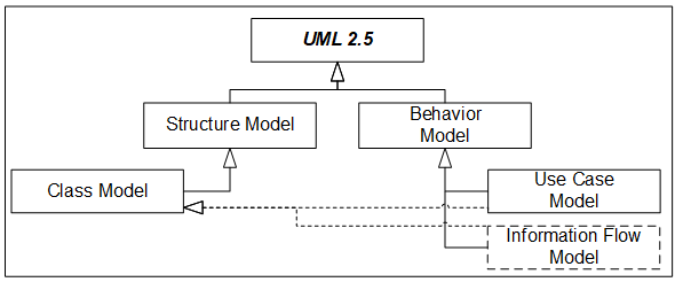


Fig. 4. *UML 2.5 Models Overview fragment [17, 21]*

Information flows can be useful to describe circulation of information through a system. These flows represents aspects of models not yet completely specified or with less details.

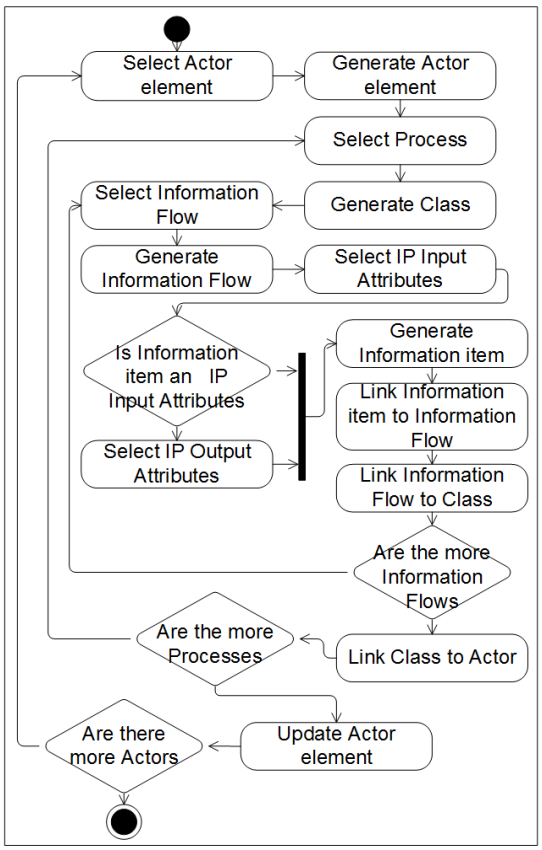


Fig. 5. *Transformation algorithm of UML Information Flow model generation from EM process*

Transformation algorithm of UML information Flow model generation from Enterprise model process is presented in the figure (Fig. 5) and is illustrated by following steps:

- Step 1: According to the top level transformation algorithm of UML models generation from EM process, UML Information Flow model is identified

for the generation process. So the initial element for UML Information Flow model is Actor element.

- Step 2: UML Information Flow model Actor element is generated from Enterprise model.
- Step 3: Process element from Enterprise model which is related with the initial actor element is selected.
- Step 4: UML Information Flow model Class element is generated from Enterprise model.
- Step 5: Information Flow element as link of other elements from Enterprise model which is related with the process element is selected.
- Step 6: UML Information Flow model Information Flow element as link of other elements is generated from Enterprise model.
- Step 7: Information processing Input Attributes element as definition of link element from Enterprise model which is related with the process element is selected.
- Step 8: If UML Information Flow model Information item element is definition of link to next element then it is generated from Enterprise model.
- Step 9: Else Information processing Output Attributes element as definition of link to previous element from Enterprise model is selected.
- Step 10: UML Information Flow model Information item element as definition of link to previous element is generated from Enterprise model.
- Step 11: UML Information flow elements Information item and Information Flow are linked.
- Step 12: UML Information flow elements Information Flow and Class are linked.
- Step 13: There is checking if there are more Information flows in Enterprise model related to UML Information Flow model. In case, there are, algorithm goes back to step 5.
- Step 14: UML Information flow elements Class and Actor are linked.
- Step 15: There is checking if there are more Processes in Enterprise model related to UML Information Flow model. In case, there are, algorithm goes back to step 3.
- Step 16: UML Information flow element Actor is updated.
- Step 17: There is checking if there are more Actors in Enterprise model related to UML Information Flow model. In case, there are, algorithm goes back to step 1.
- Step 18: Else all UML Information Flow model elements and links are generated from Enterprise Model.

B. Generated UML Information Flow Model Example

Generation of UML Information Model is illustrated with the example of Scheduled workflow for Ultrasound examination for the pet in Veterinary clinic [20, 21]. Information of this example is stored in Enterprise model. Example shows, how pet owner registers his pet in veterinary clinic to veterinary appointment in order to get the ultrasound examination, surgeon evaluation and veterinary consultation. Firstly, pet owner registers his pet in Veterinary clinic registration system, orders the ultrasound examination in ultrasound information system, then follows the process of the examination, data storage and examination data sending to surgeon, surgeon analyses examination data and writes the diagnosis using reviewing and evaluating system and sends it to veterinary through the reviewing and evaluating system, who gives the result to pet owner.

Detailed stages of Veterinary clinic example processes stored in Enterprise model are described:

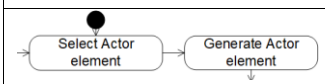
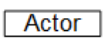
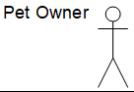
- Stage 1 – Pet owner registers his pet in veterinary clinic registration system. Information system manages pet owner registration and services ordering and is responsible for updating information.
- Stage 2 – Pet registration information from veterinary clinics registration system is connected to ultrasound examination registration system. System manages examination order scheduling.
- Stage 3 – Data gaining system acquires and creates medical data while a pet is present (in example: ultrasound, tomography etc.)
- Stage 4 – Data storage system manages examination data storage and sharing inside Veterinary clinic.
- Stage 5 – Surgeon gets data from data storage system, evaluates it through reviewing and evaluating system and prepares diagnosis response.
- Stage 6 – Veterinary gets diagnosis response prepared by surgeon through reviewing and evaluating system.
- Stage 7 – Pet owner gets diagnosis information during the appointment with veterinary.

Transformation algorithm of UML Information Flow model generation of stage 1 of Scheduled workflow for Ultrasound examination for the pet in Veterinary clinic example from Enterprise model process is illustrated by following steps:

- Step 1: Selected initial element for UML Information Flow model is Actor element.
- Step 2: UML Information Flow model Actor element is generated from Enterprise model, in certain example first actor is Pet owner.

First two steps of transformation algorithm is presented in table 4.

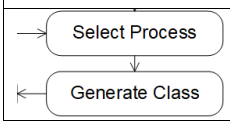
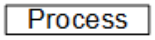
TABLE IV. STEP 1 AND STEP 2 IN UML INFORMATION FLOW MODEL GENERATION PROCESS

Transformation algorithm part	**Enterprise model element**	**Generated UML Information Flow model element**
		

- Step 3: Process element from Enterprise model which is related with the initial actor element is selected.
- Step 4: UML Information Flow model Class element is generated from Enterprise model, in certain example first class is Pet registration.

Other two steps of transformation algorithm is presented in table 5.

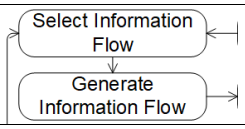
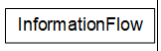
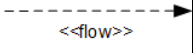
TABLE V. STEP 3 AND STEP 4 IN UML INFORMATION FLOW MODEL GENERATION PROCESS

Transformation algorithm part	Enterprise model element	Generated UML Information Flow model element
		

- Step 5: Information Flow element as link of other elements from Enterprise model which is related with the process element is selected.
- Step 6: UML Information Flow model Information Flow element as link of other elements is generated from Enterprise model, in certain example first Information flow is between Pet owner and Pet registration.

Other two steps of transformation algorithm is presented in table 6.

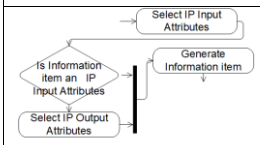
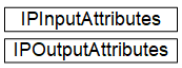
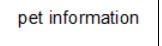
TABLE VI. STEP 5 AND STEP 6 IN UML INFORMATION FLOW MODEL GENERATION PROCESS

Transformation algorithm part	Enterprise model element	Generated UML Information Flow model element
		

- Step 7: Information processing Input Attributes element as definition of link element from Enterprise model which is related with the process element is selected.
- Step 8: If UML Information Flow model Information item element is definition of link to next element then it is generated from Enterprise model in certain example first Information item is Pet information.
- Step 9: Else Information processing Output Attributes element as definition of link to previous element from Enterprise model is selected.

Next two (in other case three) steps of transformation algorithm is presented in table 7.

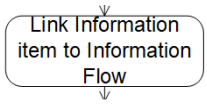
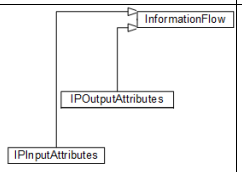
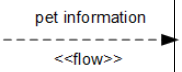
TABLE VII. STEP 7, STEP 8 AND STEP 9 IN UML INFORMATION FLOW MODEL GENERATION PROCESS

Transformation algorithm part	Enterprise model element	Generated UML Information Flow model element
		

- Step 10: UML Information Flow model Information item element as definition of link to previous element is generated from Enterprise model.
- Step 11: UML Information flow elements Information item and Information Flow are linked.

Next two steps of transformation algorithm is presented in table 8.

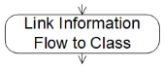
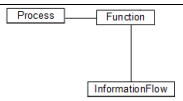
TABLE VIII. STEP 10 AND STEP 11 IN UML INFORMATION FLOW MODEL GENERATION PROCESS

Transformation algorithm part	Enterprise model element	Generated UML Information Flow model element
		

- Step 12: UML Information flow elements Information Flow and Class are linked.

Step 12 of transformation algorithm is presented in table 9.

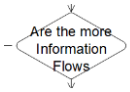
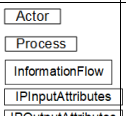
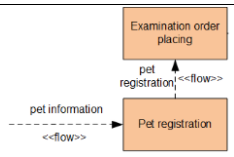
TABLE IX. STEP 9 IN UML INFORMATION FLOW MODEL GENERATION PROCESS

Transformation algorithm part	Enterprise model element	Generated UML Information Flow model element
		

- Step 13: There is checking if there are more Information flows in Enterprise model related to UML Information Flow model. In case, there are, algorithm goes back to step 5. All steps form the 5 are repeated.

Step 13 of transformation algorithm is presented in table 10, showing the result after repetition steps from step 5.

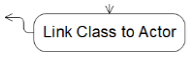
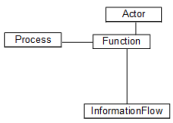
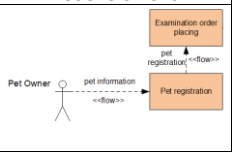
TABLE X. STEP 13 IN UML INFORMATION FLOW MODEL GENERATION PROCESS

Transformation algorithm part	Enterprise model element	Generated UML Information Flow model element
		

- Step 14: UML Information flow elements Class and Actor are linked, in certain example Pet owner is linked to pet registration.

Step 14 of transformation algorithm is presented in table 11, showing the result after repetition steps from step 5.

TABLE XI. STEP 14 IN UML INFORMATION FLOW MODEL GENERATION PROCESS

Transformation algorithm part	Enterprise model element	Generated UML Information Flow model element
		

After 14 steps of the transformation algorithm generating of Scheduled workflow for Ultrasound examination for the pet in Veterinary clinic data from Enterprise model the 1 stage – pet owner registers his pet in veterinary clinic registration system. Information system manages pet owner registration and services ordering, is responsible for updating information – is shown in the figure.

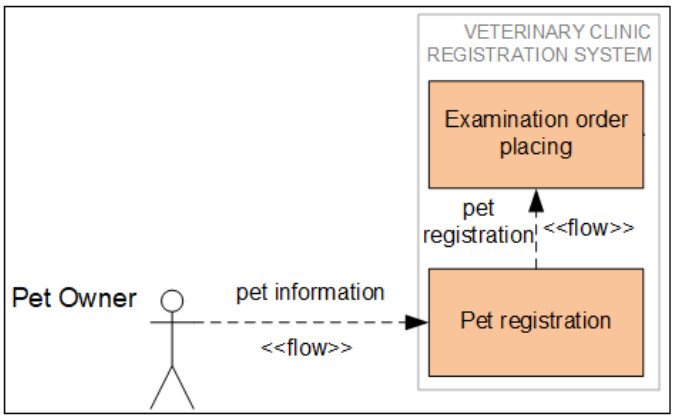


Fig. 6. *1 stage of Scheduled workflow for Ultrasound examination for the pet in Veterinary clinic example is presented as UML Information model generated from Enterprise model.*

Full UML Information flow model after all steps of the transformation algorithm generating Scheduled workflow for Ultrasound examination for the pet in Veterinary clinic example is shown in the figure.

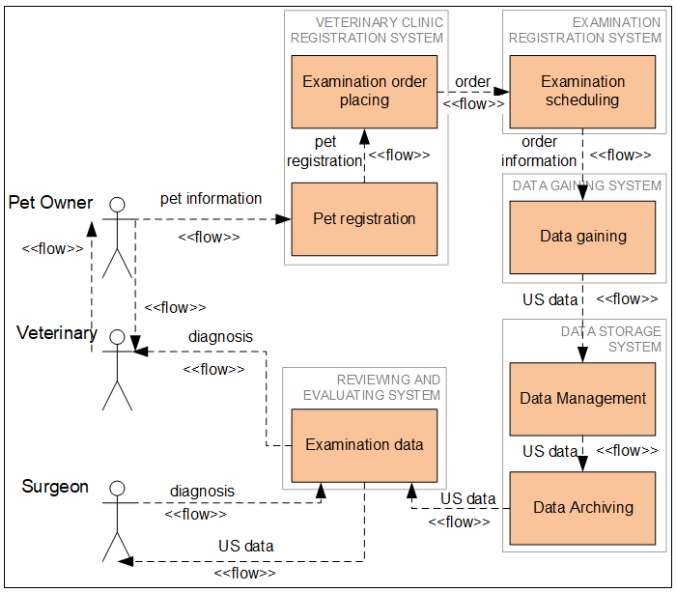


Fig. 7. *Full UML Information model genreated from Enterprise model of Scheduled workflow for Ultrasound examination for the pet in Veterinary clinic*

After the implementation all the steps of transformation algorithm it can be undoubtedly declared that chosen example perfectly illustrates accuracy of the UML Information flow elements generated from Enterprise model.

IV. CONCLUSIONS

In the first part of the article the Enterprise model elements role variations possibilities in UML dynamic models generating process and top level of transformation algorithm are presented.

The next part handles with detailed explanation of UML Information model transformation algorithm, which is depicted by steps.

In the next part there is presented particular example, which data is stored in knowledge based Enterprise model and there are described all the stages of the example.

Final part describes transformation algorithm steps for the UML Information Flow model generation from The Enterprise model and illustrates it with graphical schemes.

The illustrated example shows that data stored in Enterprise model is sufficient for generating process and it is possible to claim, that every element of UML dynamic models can be generated from the Enterprise model using transformation algorithms and this can accomplish knowledge based IS development cycle design phase.

REFERENCES

[1] Butleris, R., Lopata, A., Ambraziunas, M., Veitaitė, I., Masteika S. (2015) 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

[2] Čyras, Vytautas. (2017) Intelektualios sistemos. Vilnius: Vilniaus universitetas, Matematikos ir informatikos fakultetas, Programų sistemų katedra. 116 p. Kenneth D. Evensen and Dr. Kathryn Anne Weiss (2010)

A Comparison and Evaluation of Real-Time Software Systems Modeling Languages. AIAA Infotech@Aerospace 2010 20 - 22 April 2010, Atlanta, Georgia.

[3] IEEE Computer Society (2014) Guide to the Software Engineering Body of Knowledge SWEBOK. Version 3.0. Paperback ISBN-13: 978-0-7695-5166-1

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

[5] 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).ISBN -13: 978-1-9934272-61-9 (Volume III)

[6] Gudas S., (2012) Informacijos sistemų inžinerijos teorijos pagrindai. Vilniaus universiteto leidykla ISBN 978-609-459-075-7

[7] Gudas s., Lopata A., (2007) Meta-Model Based Development Of Use Case Model For Business Function. Information Technology And Control, ISSN 1392 – 124X 2007, Vol.36, No.3

[8] Küster, T., Lützenberger, M., Heßler, A., Hirsch, B. (2012) Integrating process modelling into multi-agent sysmtem engineering. Multiagent and Grid Systems: https://www.researchgate.net/publication/220535349_Integrating_process_modelling_into_multi-agent_system_engineering.

[9] Linehana Mark H., Barkmeyerb Ed, and Hendryx Stan, (2012) The Date-Time Vocabulary. OASIS Advancing open standards for the information society. FOIS. p. 265-378.

[10] Lopata A. Disertacija. Veiklos modeliu grindžiamas kompiuterizuotas funkcinių vartotojo reikalavimų specifikavimo metodas. 2004

[11] Lopata, A., Veitaite, I., Gudas, S., Butleris, R. (2014) CASE Tool Component – Knowledge-based Subsystem. UML Diagrams Generation Process. Transformations in Business & Economics, Vol. 13, No 2B (32B) pp. 676-696 ISSN: 1648 - 4460

[12] Lopata, A., Veitaite, I. (2013) UML Diagrams Generation Process by Using Knowledge-Based Subsystem. Tarptautinė konferencija „15th International Conference on Business Information Systems“, BIS2013 (5th Workshop on Applications of Knowledge-Based Technologies in Business (AKTB 2013)).

[13] 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.

[14] Lopata A., Ambraziūnas M., Gudas S. (2012) “Knowledge Based MDA Requirements Specification and Validation Technique”, Transformations in Business & Economics, Vol. 11, No. 1 (25), pp. 248-261.

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

[16] Morkevicius A., Gudas S. (2011) “Enterprise Knowledge Based Software Requirements Elicitation”, Information Technology and Control, Vol. 40, No 3, pp. 181-190, 1392 – 124X

[17] OMG UML (2018) Unified Modelling Language version 2.5. Unified Modelling// http://www.omg.org/spec/UML/2.5

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

[19] Veitaitė I., Lopata A. (2015) Additional Knowledge Based MOF Architecture Layer for UML Models Generation Process. 18th International Conference on Business Information Systems, BIS2015 International Workshop, Series: Lecture Notes in Business Information Processing.

[20] Veitaite I., Ambraziunas M., Lopata A. (2014) Enterprise Model and ISO Standards Based Informations System’s Development Process. 16th International Conference on Business Information Systems, BIS2014 International Workshop, Larnaca, Cyprus, May 22-23, 2014, Series: Lecture Notes in Business Information Processing.

[21] UML diagrams (2018) https://www.uml-diagrams.org/

[22] White, Stephen A. (2004) Process Modeling Notations and Workflow Patterns. IBM Corporation. p. 23-24.

[23] Zhou Xiaohua, Zhou Nan, Auto-generation of Class Diagram from Free-text Functional Specifications and Domain Ontology [Cite Seer X Beta], http://citeseerx.ist.psu.edu/