

# Additional Knowledge Based MOF Architecture Layer for UML Models Generation Process

Iлона Veitaite<sup>(✉)</sup> and Audrius Lopata

Vilnius University, Kaunas Faculty of Humanities, Muitines g. 8, Kaunas, Lithuania  
{Iлона.Veitaite,Audrius.Lopata}@khf.vu.lt

**Abstract.** As organizations grow and IS get bigger it is needed to create accurate and complicated analyzes for developing systems. This analyzes are base for design of business models and system architecture. Designers every day face new challenges when they need to collide and understand other designers' and analytics' made models. This process creates new problems and mistakes. Automization of IS engineering process lets create better and more qualified models with less mistakes. The article is about MOF architecture's role in IS engineering process and about possibilities to improve MOF architecture's composition with new knowledge based layer. The main scope of the article is to analyze improved MOF architecture's usage in UML models generation with transformation algorithms from enterprise model process.

**Keywords:** MOF architecture · Knowledge based IS engineering · Enterprise model · UML

## 1 Introduction

Nowadays information systems description with textual information and basic schemes is not enough, because of complexity of them. Modern information systems engage a variety of data, information, and knowledge-based problems. At that time, most information systems were data-oriented only. Their initial purpose was to store, retrieve and control data [1–4].

Information system engineering extends through the whole life cycle of systems. Information system engineering is based on the traditional knowledge and empirical experience of the analyst combined with supplementary abilities achieved from previous practice. Information systems engineering process is a logical sequence of actions and solutions that converts operational demands into a description of system performance configuration [5, 6].

The Meta-Object Facility (MOF) is an Object Management Group (OMG) standard for model-driven engineering. Its purpose is to provide a type system for entities and a set of interfaces through which those types can be created and manipulated. MOF only provides a means to define the structure, or abstract syntax of a language or of data [7].

Enterprise modeling has become an inextricable part of information system development process. Recently, the organization's business modeling has become an important

phase of modeling design processes. MOF architecture supplemented with enterprise meta-model assures appropriate information system development process.

## 2 MOF Architecture

Computerized IS engineering is developing based on new knowledge-based methods, in order to avoid empirical influence. A knowledge-based IS engineering offers system modeling and decision-making methods and tools, which helps to develop more precise and detailed subject area corresponding to the project.

The Meta-Object Facility (MOF) is an Object Management Group (OMG) standard is designed as a four-layered architecture. It provides a meta-meta model at the top layer, called the M3 layer. This M3-model is the language used by MOF to build meta-models, called M2-models. The most prominent example of a Layer 2 MOF model is the UML meta-model, the model that describes the UML itself. These M2-models describe elements of the M1-layer, and thus M1-models, for example, models written in UML. The last layer is the M0-layer or data layer. It is used to describe real-world objects. MOF is a closed meta-modeling architecture; it defines an M3-model, which conforms to it-self [7] (Fig. 1).

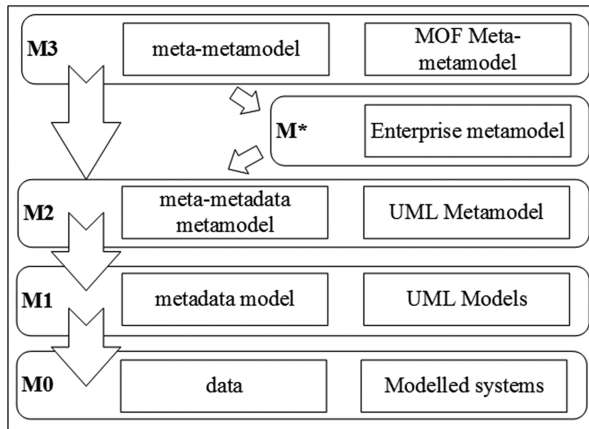


Fig. 1. MOF architecture with additional EM layer

As it is described above M3 is meta-meta model, the base for a Metamodeling Architecture, which defines the language to describe meta-models. And M2 meta-model is an instance of a meta-meta model, which defines the language to describe models. Among M3 and M2 layers one more layer is needed to assure more accurate usage of MOF architecture [7]. This additional layer consists of enterprise meta-model (EMM). Enterprise meta-model is a formal structure which ensures more qualified information system development process and knowledge base data collection. Enterprise model and enterprise meta-model makes information system needed UML project models generation process more efficient and eligible and assure reduced number of mistakes in the final information system development phase [1].

### 3 Enterprise Model

Enterprise meta-model is formally defined enterprise model structure, which consists of a formalized enterprise model in line with the general principles of control theory. Enterprise model is the main source of the necessary knowledge of the particular problem domain for IS engineering and IS re-engineering processes [8].

Enterprise meta-model manages Enterprise model composition and stores knowledge that is necessary for information system development process only. It can be used during all phases of information system development life cycle [9, 10] (Fig. 2).

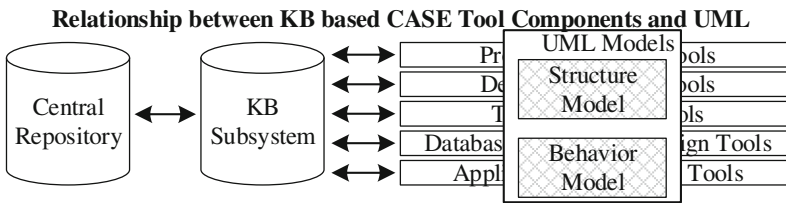


Fig. 2. Relationship between KB based CASE tool components and UML

Enterprise model and Enterprise meta-model are the main components of knowledge-based subsystem of CASE Tool. It is suitable to collect necessary knowledge to the knowledge-based subsystem of the particular CASE Tool. This subsystem is the main source of knowledge necessary for design phase models, for example UML models through transformation algorithms, and source code generation process [10, 11].

There is given formalized Enterprise meta-model description, which is needed to define UML Sequence model generation process algorithm. Enterprise model can be described as Malcev algebra based algebra system [12]:

$$M1 = \langle K, R \rangle ; \tag{1}$$

- 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:

$$Kn = \langle \{an1, an2, \dots, ank\}, \{mn1, mn2, \dots, mnl\} \rangle , \text{ where}$$

$$\{an1, an2, \dots, ank\} - \text{attributes of } Kn \text{ element,}$$

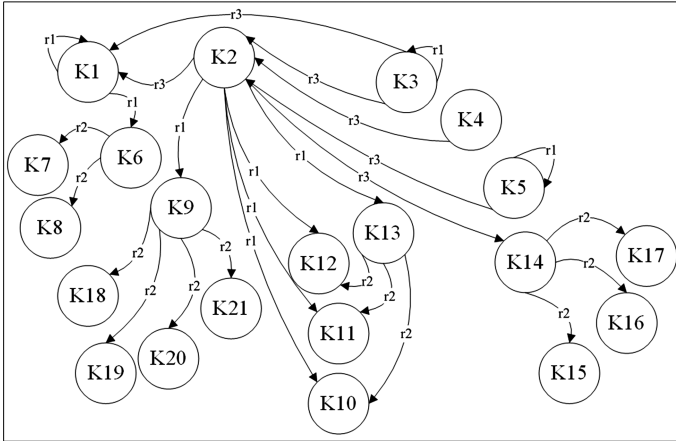
$$\{mn1, mn2, \dots, mnl\} - \text{methods of } Kn \text{ element.}$$

Enterprise model M1 composition is as follows:

$$M1 = \langle \{K1, k2, \dots, K21\}, \{r1, r2, r3\} \rangle ; \tag{2}$$

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, K20 – meta-class Information processing Output, K21 – meta-class Process Input, r1 – Aggregation, r2 – Generalization, r3 – Association. Enterprise meta-model graphical schema is presented at Fig. 3:



**Fig. 3.** Enterprise meta-model graphical schema based on Malcev algebra

## 4 UML Models Generation Process

UML model is a partial graphical view of a model of a system under design, implementation, or already in presence. UML model contains graphical elements – UML nodes connected with flows – that represent elements in the UML model of the designed system [13].

The kind of the model is described by the primary graphical symbols shown on the model. UML specification does not prevent mixing of different kinds of models, for example to unite structural and behavioral elements to express particular case. Therefore, the boundaries between the various kinds of models are not strictly enforced [14].

UML specification defines two kinds of UML models: structure models and behavior models, where Structure models show the static structure of the system and its parts on different abstraction and implementation levels and how they are related to each other. The elements in a structure model represent the meaningful concepts of a system, and may include abstract, real world and implementation concepts; and Behavior models show the dynamic behavior of the objects in a system, which can be described as a series of changes to the system over time [13, 14].

One of the behavior models is Sequence model. It is most common kind of interaction models which focuses on the message interchange between objects (lifelines). Sequence model shows how the objects interact with others in a particular scenario of a use case [14].

There is given formalized UML Sequence model description. UML Sequence model also can be described as Malcev algebra based algebra system [12]:

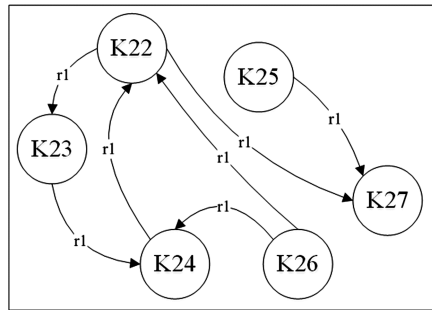
$$M2 = \langle K, R \rangle ; \tag{3}$$

where  $M2$  – UML Sequence model as algebra system;  
 $K$  – elements set of  $M2$  system  
 $K = \{K22, K23, \dots, K27\}$ , where  $K22, \dots, K27$  UML Sequence model meta-classes  
 $R$  – set of relationships between elements, where  $R = \{r1, r2, r3\}$ .

UML Sequence Model  $M2$  composition is as follows:

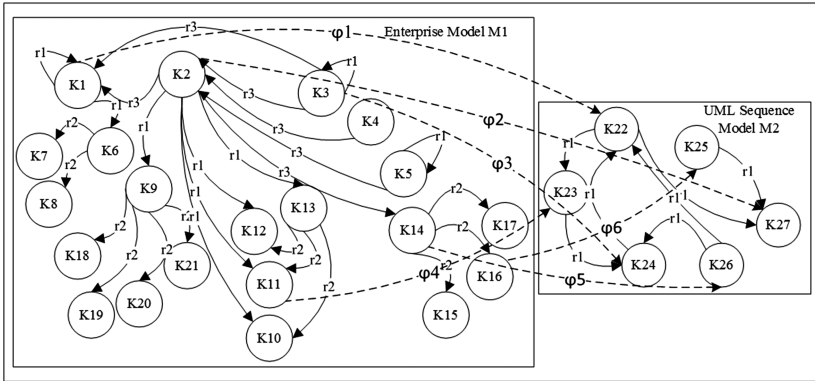
$$M = \langle \{K22, K23, \dots, K27\}, \{r1\} \rangle ; \tag{4}$$

where:  $K22$ – meta-class Message,  $K23$  – meta-class Execution,  $K24$  – meta-class Life-line,  $K25$ – meta-class State Invariant,  $K26$  – meta-class Combined Fragment,  $K27$  – meta-class Destruction Occurrence,  $r1$  – Aggregation. UML Sequence Model graphical schema is presented at Fig. 4:



**Fig. 4.** UML sequence model graphical schema based on Malcev algebra

Intersection between Enterprise model and UML Sequence model elements is presented in the Table 1 and in graphical scheme in Fig. 5.



**Fig. 5.** Graphical scheme of intersection between enterprise model and UML sequence model elements based on Malcev algebra

Transformation algorithm presented in the Fig. 6 is algorithm for enterprise meta-model based UML Sequence model generating process. Main steps for generating process are identifying and selecting UML Sequence model for generating process, identifying starting elements, selecting elements types, selecting relationships between elements types, selecting all related elements, reflecting enterprise model elements to UML model elements and generating selected UML sequence model elements.

**Table 1.** Intersection between enterprise model and UML sequence model elements

Enterprise model set element	UML sequence model set element	Formal description
Process (K1)	Message (K22)	$\varphi1: K1 \rightarrow K22$
Function (K2)	Destruction occurrence (K27)	$\varphi2: K2 \rightarrow K27$
Actor (K3)	Lifeline (K24)	$\varphi3: K3 \rightarrow K24$
Data processing and solution making (K11)	Execution (K23)	$\varphi4: K11 \rightarrow K23$
Business rules (K14)	Combined fragment (K26)	$\varphi5: K14 \rightarrow K26$
Data processing and solution making business rules (K16)	State invariant (K25)	$\varphi6: K16 \rightarrow K25$

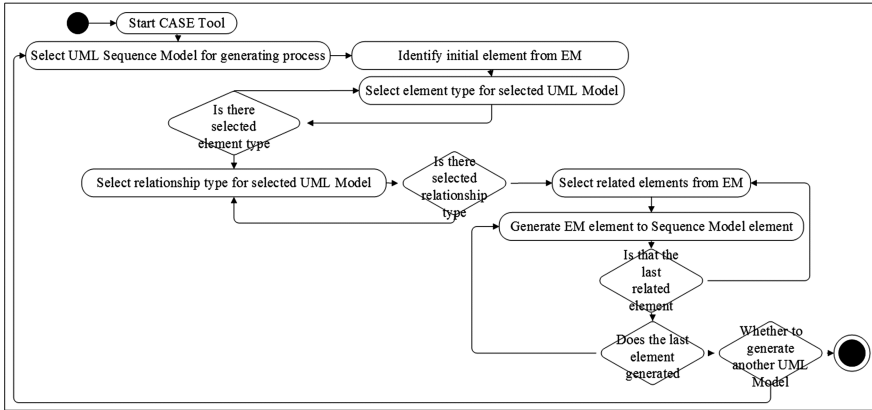


Fig. 6. UML sequence model generation from enterprise model algorithm

## 5 Conclusions

Qualitatively implemented information system development life cycle phases are very significant for the success of information systems development process and mistakes made in that phases will cause huge problems and cost a lot of time and expenses to solve it.

MOF architecture was established to provide a suit of schemas by which the structure, meaning and behavior of objects could be described. Enterprise meta-model as an additional MOF layer ensures that enterprise model includes business management information process fundamental attributes and provides a knowledge base as a CASE Tool subsystem, which ensures quality and verified knowledge in specific scenarios. Each element of UML models can be generated from the enterprise model using knowledge based enterprise model and transformation algorithms.

MOF architecture with additional layer of enterprise meta-model simplifies UML models generation process from knowledge-based enterprise model, IS development process becomes consistent, more effective and there is lower number of mistakes in the closing information system development phase.

## References

1. Butleris, R., Lopata, A., Ambraziunas, M., Veitaitė, I., Masteika, S.: SysML and UML models usage in knowledge based MDA process. *Elektronika ir elektrotechnika*. **21**(2), 50–57 (2015). Print ISSN: 1392-1215, Online ISSN: 2029-5731
2. Eichelberger, H., Eldogan, Y., Schmid, K.: A comprehensive analysis of UML tools, their capabilities and compliance. In: *Software Systems Engineering, V.2.0*. Universität Hildesheim (2011)
3. Jenney, J.: *Modern methods of systems engineering: with an introduction to pattern and model based methods* (2010). ISBN-13:978-1463777357

4. Perjons, E.: Model-driven process design. aligning value networks, enterprise goals, services and IT systems. Department of Computer and Systems Sciences, Stockholm University. Sweden by US-AB, Stockholm (2011). ISBN 978-91-7447-249-3
5. Freiberg, M., Baumeister, J., Puppe, F.: Interaction pattern categories — pragmatic engineering of knowledge-based systems. In: 6th Workshop on Knowledge Engineering and Software Engineering (KESE6) Germany (2010)
6. Ulrich, W.M.: Information Systems Transformation: Architecture-Driven Modernization Case Studies. Morgan Kaufmann, Newcomb Amsterdam (2010). Elsevier
7. OMG MOF.: OMG Meta Object Facility (MOF) Core Specification, Version 2.4.2, April 2014 (2015). <http://www.omg.org/spec/MOF/2.4.2>
8. Gudas, S.: Architecture of knowledge-based enterprise management systems: a control view. In: Proceedings of the 13th World Multiconference on Systemics, Cybernetics And Informatics, Orlando, Florida, USA, vol. III, p.161–266 (2009). ISBN -10: 1-9934272-61-2
9. Gudas, S.: Informacijos sistemų inžinerijos teorijos pagrindai, p. 384. Vilnius University (2012). ISBN 978-609-459-075-7
10. Lopata, A., Ambraziūnas, M., Gudas, S., Butleris, R.: The main principles of knowledge-based information systems engineering. Electron. Electr. Eng. **11**(1(25)), 99–102 (2012). ISSN 2029-5731
11. Veitaite, I., Lopata, A., Gudas, S., Butleris, R.: CASE tool component – knowledge-based subsystem uml diagrams generation process. Transf. Bus. Econ. **13**(2B(32B)), 676–696 (2014). ISSN: 1648 – 4460
12. Gudas, S., Lopata, A.: Žiniomis grindžiama sistemų inžinerija, Mokomoji medžiaga. Vilnius University, Kaunas Faculty of Humanities (2011)
13. UML Diagrams: Diagrams characteristics (2015). [www.uml-diagrams.org](http://www.uml-diagrams.org)
14. OMG UML.: Unified Modeling Language version 2.5. Unified Modeling (2015). <http://www.omg.org/spec/UML/2.5/Beta2/>