

The Main Principles of Knowledge-Based Information Systems Engineering

A. Lopata, M. Ambraziunas, S. Gudas

Kaunas Faculty of Humanities, Vilnius University,

Muitinės St. 8, LT-44280 Kaunas, Lithuania, phone +370 37 422566, e-mails: audrius.lopata@khf.vu.lt,

martas.ambraziunas@khf.vu.lt, saulius.gudas@khf.vu.lt

R. Butleris

Department of Information Systems, Faculty of Informatics, Kaunas University of Technology,

Studentu str. 50-313a, LT-51368 Kaunas, Lithuania, phone: +370-37-453445, e-mail: rimantas.butleris@ktu.lt

crossref <http://dx.doi.org/10.5755/j01.eee.120.4.1192>

Introduction

One of the most important issues of Information Systems Engineering (ISE) is software validation against user requirements. The core of this issue is as follows: expert plays the main role in the problem domain knowledge acquisition and analysis process, therefore user requirement analysis and enterprise modelling process heavily depend on the expertise of system analyst and the user [1]. In order to solve this issue, the main principles of Knowledge-Based ISE could be used. The core idea of Knowledge-Based ISE is as follows: problem domain knowledge is validated against formal criteria and stored in Knowledge-Based subsystem [2].

Knowledge-Based MDA combines two software development approaches: Knowledge-Based IS Engineering (KBE) and Model Driven IS Engineering (MDE). The core element of KBE is Enterprise meta-model. There are many different enterprise modelling standards including: CEN EN 12204, CEN EN 40003 (CIMOSA), UEML. These standards have the following shortcomings in ISE user requirements acquisition, analysis and specification steps:

- The content of enterprise models, presently used in CASE systems, is not verified according to formalized criteria.
- Traditionally problem domain modelling and user requirements specification processes are performed by the analyst and user, who use their own experience and knowledge, thus the results, have empirical nature.

In order to solve these problems a particular Enterprise meta-model [3], has been developed by a joint scientific group of Kaunas University of Technology and

Vilnius University. Enterprise meta-model's internal structure is based on Control Theory [4] and best practices of the above mentioned enterprise modelling standards. Several authors [2], [3], [5], [6] have recently proposed EMM based UML models generation methods in order to decrease empirical nature of Information Systems Engineering process. These methods are able to cover some aspects of MDA, though the mentioned above EMM is not MDA compatible enough to fully support MDA process yet [2].

Numerous tools that partly or fully implement MDA ideology e.g. AndroMDA, Enterprise Architect, Acceleo, Magic Draw, exist in the market [7]. Most of these tools implement generation and composition functionalities and focused on PIM/PSM/Code models of MDA process and there is lack of tools that analyse CIM models and validate them against user requirements and business modelling standards.

Knowledge based MDA

Knowledge-Based MDA approach combines the basic MDE and KBE principles. The main components of this approach are presented in Fig. 1. MDE defines the main principles of domain models usage in IS development process. MDE is the basis for MDA approach. This MDE branch is defined by OMG using standards like MOF, XMI, and UML. Although MDA is less abstract approach than MDE but it still provides only guidelines (and standards) for IS development, but not a real working solution.

The next abstraction level is methods based on MDA e.g. Agile MDA as well as Knowledge-Based MDA belongs to this level.

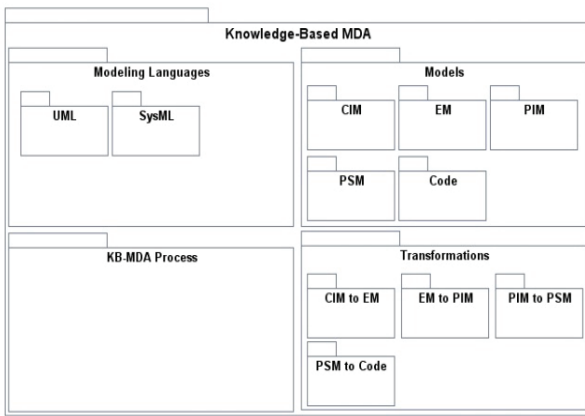


Fig. 1. The main elements of Knowledge-Based MDA

Basically Knowledge-Based MDA extends MDA with additional model (Enterprise Model) and two transformations (CIM to EM and EM to PIM).

The main activities of Knowledge-Based MDA process

Knowledge-Based MDA process involves five actors: User, System Analyst, Knowledge-Based subsystem, MDA model transformation (generation) tool, and Developer. User is participating in the first stage of IS development (CIM creation) by providing requirements and requirements related data (Fig. 2.). Based on provided information System Analyst creates UML models that represent particular requirements area i.e. Use Case model and Activity model for functional requirements, Block Definition model for Systems structure, and Requirements model for non-functional requirements. After all internal CIM models are created, syntactic and semantic models validation is performed. After validation is successfully completed, Knowledge-Based subsystem creates Enterprise Model based on existing UML models. The next stage is Enterprise Model’s validation against Enterprise Meta-Model. This process produces validation report. Validation report is analysed by System Analyst and a particular decision is made whether to proceed with the next steps or repeat the initial steps. If the decision is made to proceed, System Analyst selects a particular area for which programing code will be generated. Based on this decision the PIM and PSM models are created. From the PIM layer generation the process follows standard MDA process with one exception any changes that are made to models must be synchronized with Enterprise Model. We will look at all Knowledge-Based MDA activities in a more detailed way in the next chapters.

CIM layer process

The main steps of CIM layer process are presented in Fig. 2. The first step is “Provide information about system requirements”. At this step user describes necessary system’s behaviour and scenarios (functional requirements). System Analyst creates a set of Use Case models to capture these requirements. Based on created Use Case models and captured user scenarios Activity models are created. After this step CIM model contains two internal models that are representing system’s

behaviour. In step “Provide information about system’s structure” User describes compositional parts of System. Analyst captures this data by using Block Definition diagrams. After system’s behaviour and structure are captured one more step involving requirements is performed – non- functional requirements are captured by using Requirements models.

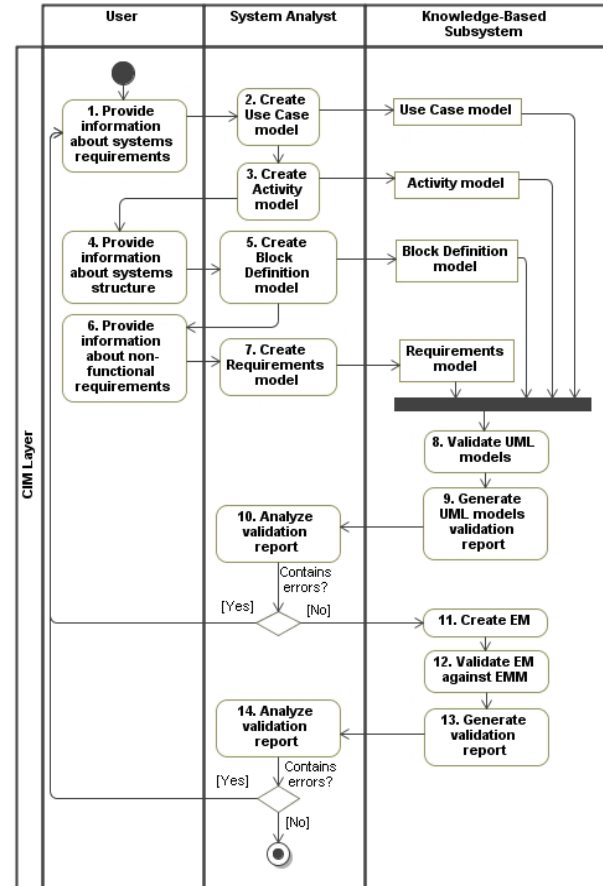


Fig. 2. The main steps of CIM layer process

These requirements are associated with Use Case models. At step No. 8 “Validate UML models” Internal CIM models are validated against each other. The main purpose of this step is to resolve any issues among models e.g. missing or duplicated elements, inconsistent elements, missing flows etc. Next step No.9 is “Generate UML models validation report”. System Analyst reviews validation report in step No.10 “Analyse validation report”, based on this report he decides what action should be taken (basically there are two options: to proceed with the next steps or repeat requirements acquisition process in order to fix the existing errors and issues). If there are no errors or System Analyst decides to ignore them (no major issues exists), Enterprise Model is created from UML models. EM is validated against Enterprise Meta-Model at step No. 12 “Validate EM against EMM”. If validation is completed successfully the next steps can be performed, in other case requirements acquisition process should be repeated. This EM validation/requirements acquisition process is iterative and is repeated while EM is correct against EMM formal criteria. After validation is completed successfully next stages of KB-MDA process can proceed (PIM and PSM

layers).

PIM and PSM layer process

The main steps of PIM and PSM layer processes are presented in Fig. 3. This process starts from step No.15 “Select Area”. In some cases there could be no reason to create all IS elements. Instead it could be useful to create only some parts of application (Information System can be created and deployed by parts or modules). At this step the particular problem domain area is selected, for which new IS elements will be created. After the particular area is selected, the predefined UML models (PIM internal models) are created from Enterprise Mode: “Class models” (static), “Sequence models” (behaviour), “Use Case model” (behaviour).

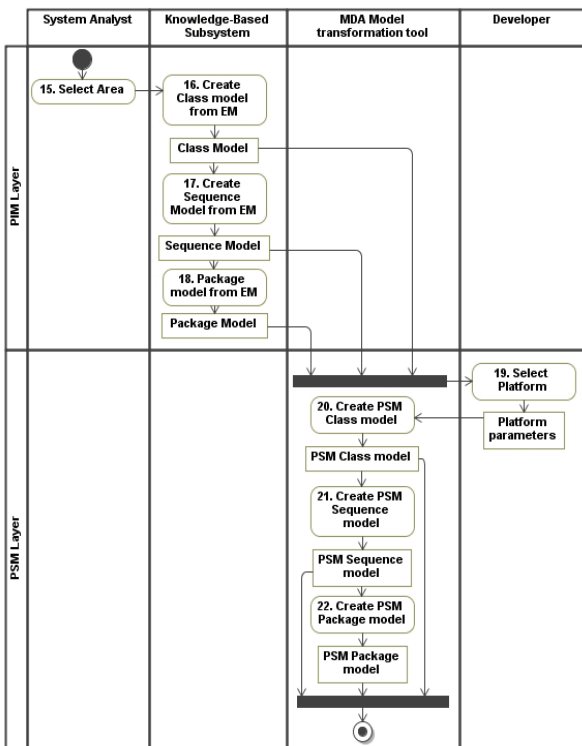


Fig. 3. The main steps of PIM and PSM layer process

These models don't have any platform specific information (represents basic system behaviour and structure) and are used for PSM models creation (traditional MDA). Before PSM models could be created, platform selection should be made. This action is performed at step No.19 “Select Platform”. Platform selection is made by Developer (Architect/Lead Developer). After platform selection PIM model is transformed to PSM model by appending it with platform specific tags and elements.

PSM contains the same type of internal UML models, but with different abstraction level. These models contain platform specific elements e.g. data types, parameters descriptions etc. When PSM models are generated the next steps (Code Layer) can be preceded.

Code layer process

The main steps of Code layer process are

presented in Fig. 4. Code layer generation process is based on three-layer applications architecture: data layer, business layer, graphic user interface layer. This process starts from step No.23 “Data Layer Code generation”. Data Layer code is generated from class models. Depending on the selected data layer platform code could be generated to SQL, XML schemas or other standards (if there are defined mappings with class model).

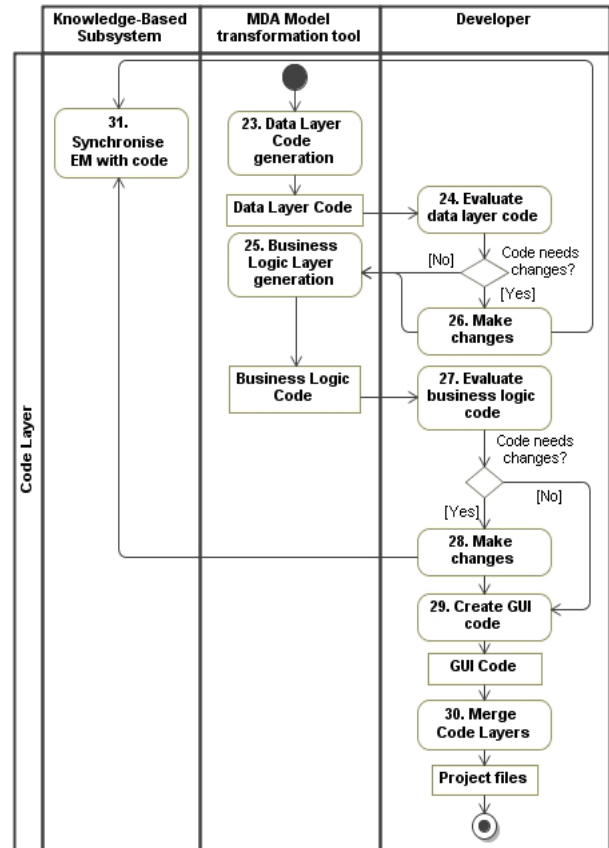


Fig. 4. The main steps of Code layer process

Class models describe business objects, not Data Layer object, so optimization algorithms (e.g. DB normalization) should be applied to create adequate Data Layer code or use Data Layer developer's reviews. If data Layers Code that was generated needs some changes that are provided by developer, the updated code should be synchronized with EM. The next layer to be generated is Business Logic layer. This is performed at step No.25 “Business Logic Layer generation”. This layer is generated from Class, Package and Sequence models. Class models provide framework for business objects, Package models provides grouping framework for these objects and sequence models provide behaviour specifications for these objects. Each business object should be used in one or more sequence diagrams. Business Logic code is evaluated by the developer. If it is necessary developer can make changes to code. After changes are made synchronization with EM should be performed. After Data Layer and Business Logic Layer are created, GUI Layer (if it is necessary) is created from specific Class model [6] or by developer. The GUI generation from UML models can be performed but this process needs developer's involvement depending on

systems purpose. Applications for which GUI is not important (e.g. console based applications) can be deployed without modifications, but for applications that need to implement sophisticated GUI this step cannot be made fully automatically. After all three code layers are created and synchronized with Enterprise Model, the code can be merged in a single project file (files). Merge is performed by developer using selected IDE. This step is the final step of development process and the final result of it is project file in some IDE (e.g. Visual Studio) format that can be compiled. Deployed system can be tested using scenarios that are created based on Use Case and Requirements models. In ideal scenario unit testing should be used in most cases. GUI testing is performed manually by the testers. The final step of KB-MDA process is systems deployment to working environment.

Conclusions

The Main Principles of Knowledge-Based Information Systems Engineering is presented in the article. The set of four types of UML and SysML models like Use Case Model, Activity Model, Block Definition Model and Requirements Model are necessary in order to acquire problem domain knowledge completely. The knowledge-based subsystem is additional actor which ensures that information validation and verification processes according to user requirements using formal criteria will be performed completely and successfully. The proposed Model Driven Architecture compatible

method is designed by using the following technologies: modelling languages UML and SysML as well as developed using Object Oriented programming languages (C#), Relational Database (SQL Server) and XML (XMI) as data transferring standard.

References

1. **Vernadat F.** UEML: Towards a Unified Enterprise modeling language // *International Journal of Production Research*, 2002. – Vol. 40. – No. 17. –P. 4309–4321.
2. **Lopata A., Ambraziūnas M.** Knowledge-Based MDA Approach // *BIS 2011 International Workshops.*– Poznan, Poland, 2011.– Series: Lecture Notes in Business Information Processing. – Vol. 97. – P.160–165.
3. **Gudas S., Lopata A., Skersys T.** Approach to enterprise modelling for information systems engineering // *Informatica*, 2005. – No. 16(2). – P. 175–192.
4. **Gupta M. M., Sinha N. K.** Intelligent Control Systems: Theory and Applications. – IEEE Press, 1996. – 820 p.
5. **Skersys T.** Business Knowledge-Based Generation of the System Class Model // *Information technology and control*, 2008. – Vol. 37. – No. 2. – P. 145–153.
6. **Čeponienė L., Nemuraitė L., Vedrickas G.** Separation of event and constraint rules in UML&OCL models of service oriented information systems // *Information technology and control*, 2009. – Vol. 38. – No. 1. – P. 29–37.
7. **Silingas D., Pavalkis S., Vitiutinas R., Nemuraite L.** Integrating GUI Prototyping into UML Toolkit // *Proceedings of 16th International Conference “Information Technologies 2010”.* – Kaunas, Lithuania, 2010. – P. 224–233.

Received 2011 11 16

Accepted after revision 2012 01 23

A. Lopata, M. Ambraziunas, S. Gudas, R. Butleris. The Main Principles of Knowledge-Based Information Systems Engineering // *Electronics and Electrical Engineering.* – Kaunas: Technologija, 2012. – No. 4(120). – P. 99–102.

This survey proposes Knowledge-Based MDA method which realizes MDA methodology and appends it with Knowledge-based validation of CIM model has been proposed in the article. This validation is performed against the formal criteria that are described in Enterprise meta-model (formal structure that defines how an organization should work in ideal case). Applications created using this method could ensure the provided functionality highly conforms to user needs. The proposed method will be realized by using the following technologies: modeling languages UML and SysML, Object Oriented programming languages (C#), Relational Database (SQL Server) and XML (XMI) as data transferring standard. Ill. 4, bibl. 7 (in English; abstracts in English and Lithuanian).

A. Lopata, M. Ambraziūnas, S. Gudas, R. Butleris. Pagrindiniai žinios grindžiamos informacijos sistemų inžinerijos principai // *Elektronika ir elektrotechnika.* – Kaunas: Technologija, 2012. – Nr. 4(120). – P. 99–102.

Šiame straipsnyje pateiktu tyrimu siekiama sukurti informacijos sistemų inžinerijos metodą, užtikrinantį dalykinės srities modeliavimo, vartotojo reikalavimų specifikuojimo ir projektavimo informacijos sistemų (IS) inžinerijos etapų procesų integraciją, jungiant žinios grindžiamos IS inžinerijos principus su vaizdiniais modeliais grindžiamos IS inžinerijos (MDA) principais. Sėkminga šių metodų integracija įgalina sukurti kokybiškai naują IS kūrimo metodą, pasižymintį didesniu lankstumu bei mažiau žmogiškųjų, laiko ir finansinių išteklių reikalaujančiu IS kūrimo procesu. Siūlomas metodas įgyvendinamas taikant veiklos modeliavimo kalbas (UML ir SysML), objektiškai orientuotą programavimo kalbą (C #), reliacinę duomenų bazę ir duomenų perdavimo standartą XML (XMI). Il. 4, bibl. 7 (anglų kalba; santraukos anglų ir lietuvių k.).