Principles of Model Driven Architecture for the task of study program development

O. Nikiforova¹, U. Sukovskis², V. Nikulsins²

¹Faculty of Computer Science and Information Technology, Riga Technical University, LV-1658 Riga, Latvia (oksana.nikiforova@rtu.lv)
²Faculty of Computer Science and Information Technology, Riga Technical University, LV-1658 Riga, Latvia

Abstract

The Model Driven Architecture (MDA) separates the system business aspects from the system implementation aspects on a specific technology platform. MDA proposes a software development process in which the key notions are models and model transformation, where the input models are platform independent and the output models are platform specific and can be transformed into a format that is executable.

In this paper we apply principles of MDA for the task of study program development. Each study program may be considered as a “platform” specific knowledge, because it is delivered by means of particular courses, which are taught by particular teachers using particular study equipment. The principles of MDA for the task of study program development are applied using Two-hemisphere model driven (2HMD) approach, which assumes modelling and use of procedural and conceptual knowledge and may be applied in the context of modelling the knowledge about the domain.

From the point of view of 2HMD approach each course of the study program may be considered as a knowledge provider, which satisfies particular knowledge requirements. The particular “specific” knowledge requirements are derived from the knowledge model that consists of functional and conceptual “hemispheres”, which reflect knowledge derived from original academic or industrial requirements or standards.

Keywords: model driven architecture, two-hemisphere model driven approach knowledge modelling, study program, conceptual modelling, business process modelling

1. INTRODUCTION

Present Latvian higher educational system is not able to react quickly enough the industrial requirements. The use of new technologies and modern trends are not being considered sufficiently. One of the possible solutions is a regular study program review and integration of new modules into the current educational program, preserving the study structure and the balance of different disciplines. The goal of this paper is to propose a new methodological approach integrating new study modules into the current study program using of Two-hemisphere model driven approach and principles of Model Driven Architecture.

Two-hemisphere model driven (2HMD) approach has been successfully applied for modelling and software design domain [1], [2]. One of the most distinguished features of this model is its applicability for both human understanding and automatic transformations [3], [4]. In this paper we illustrate the way how 2HMD approach can be applied to the task of knowledge modelling for educational purposes. This task becomes more and more important because of urgent need of knowledge provision in the knowledge based economy. The need of providing knowledge timely and accurately arises in different settings, starting from industrial needs to training employees in new technologies and ending with universities that have to become more and more flexible in terms of their curricula and course development [5]. For example, undergraduate Information Systems Model Curriculum that was first developed in 1972 and updated in 1982, 1997 and 2002 [6]. So the last update was made two or three times shorter than the previous ones. This tendency of more frequent changes in curricula and
course contents requires special means for course and curricula maintenance in terms of knowledge modelling and provision [7]

As a practical example of such approach new study module integration into the current study program “Computer Systems” is being demonstrated and implemented in the Faculty of Computer Science and Information Technology in Riga Technical University. The need of such development is a desire to focus Latvian IT industry to the MDA (Model Driven Architecture) [8], which is being developed by OMG Inc. in cooperation with EU Scientific Frameworks programs.

2. CENTRAL IDEA OF STUDY MODULE DEVELOPMENT

The base of any profession is a theoretical and practical knowledge. It is preferable to define this knowledge for the study module development, certification of specialists, and profession licensing. The contents and form of the fast-developing fields taught to the students should be reviewed and analysed on the regular basis [9].

Software engineering is a very fast developing knowledge area, which is being affected by external factors like political, economical, historical, technical etc. This requires high level of flexibility [10], which means a study program contents change according to the external environment, and from the other side ability to persist study program invariability. Such relative invariability allows preserving legal status and avoids loss of accreditation or license for the study program [11]. Study module can be considered from the system theory view [12] (graphical representation is shown in Figure 1).

\[ Y = F (X) \]

FIGURE 1. Arguments and results of the function of study module development

3 KNOWLEDGE MODELING PRINCIPLES OF INTEGRATED KNOWLEDGE AREA

In industrial setting the following classification of knowledge has been introduced and is used to analyze company’s competitive position [13], [14]: core knowledge; advanced knowledge and innovative knowledge.

Core knowledge is the basic scope and level of knowledge that represents a basic industrial knowledge barrier to entry. Advanced knowledge enables the company to be competitively viable. It is company’s superior knowledge in certain areas. Innovative knowledge allows company to lead the entire industry.

The same classification of knowledge can be applied to the contents of curricula like [15] or [16] and courses in continuously changing knowledge areas to characterize the competitiveness of a curriculum and, consequently, the competitiveness of students in the labour market. Thus, from the point of view of curriculum, the core knowledge is knowledge that guarantees students’ ability to enter successfully the labour market, advanced knowledge provides background for advanced positions in the labour market, and innovative knowledge supports opportunities to obtain leading positions in the industrial labour market [5].

Taking into consideration knowledge partitioning given above, it is obvious that knowledge model should be developed in a manner that represents all three types of knowledge, and, on the other hand, gives an opportunity not only to distinguish between those types of knowledge, but also to move knowledge from one type to another when a particular piece of advanced knowledge becomes core knowledge and innovative knowledge becomes advanced knowledge [14]. This feature of knowledge model partly is achieved by module principle in curriculum development.

However, the knowledge is represented in the module description as traditional sentence-based expressions that allow a limited level of transparent details of knowledge representation. A higher level of detail may be achieved when course descriptions are added to the module description. However, it does not provide any formal means for consistency and completeness check of knowledge actually provided. Moreover, informally represented knowledge is hardly comparable to particular knowledge standards [5] because of differences in terms,
granularity of knowledge and variety of possible interpretations. Therefore informal descriptions do not support manageability of knowledge to be provided in a frequently changing educational environment [9].

One of the modern research goals in software engineering is to find a software development process, which would provide fast and high quality software development. Most of currently proposed methodologies and approaches try to make the development process easier and still more qualitative. For achievement of this goal the role of explicit models becomes more and more important. Lately, the most popular approach is Model Driven Architecture (MDA) [8], [17].

Similarly the goal of education in knowledge based economy is to provide fast and qualitative education. Therefore it is potentially promising to try to achieve it in a similar to software development way, i.e., by applying the principles of MDA.

MDA introduces an approach to system specification that separates the views on three different layers of abstraction: high level specification of what the system is expected to do (Computation Independent Model or CIM), the specification of system functionality (Platform Independent Model or PIM) and the specification of the implementation of that functionality on a specific technology platform (Platform Specific Model or PSM).

CIM presents specification of the system at problem domain level and can be transformed into elements of PIM. PIM provides formal specification of the system structure and functions that abstracts from technical details, and thus presents solution aspects of the system to be developed, which enables model transformation to the platform level (PSM), named implementation domain in Figure 2.

![Figure 2](image)

**FIGURE 2.** Model transformation from problem domain level of knowledge representation into implementation domain level according 2HMD approach

The details in the right column of the table in Figure 2 correspond to the 2HMD approach [1], [2], which addresses the construction of information about problem domain by use of two interrelated models at problem domain level, namely, the process model and the conceptual model. The conceptual model is used in parallel with process model to cross-examine software developers understanding of procedural and semantic aspects of problem domain.

The name of the two-hemisphere approach is a metaphor with roots in cognitive psychology. It is based on the statement about interconnected hemispheres of human brains, one of which is responsible for objects (things), and the second one is responsible for logics. Condition for adequate behaviour of human is collaboration of both hemispheres. Two-hemisphere metaphor is being used in software architecture concepts development, interpreting software systems as collaboration between functional model and conceptual model.

Model transformation in terms of diagrams based on object-oriented software development is shown in Figure 3. Two-hemisphere model, which is an interrelated combination of process model and conceptual model, serve as a basis for task of application boundary restriction, where only the processes have to be automated by developed software has to be identified for detailed definition of sub-process diagram development [1]. Then based on defined transformation algorithm and formal transformation rules from sub-process model into appropriate class structure of developed software systems, class diagram is developed and contains all the necessary encapsulation of structural features and behavioural responsibilities of problem domain among software components.

Similar transformations can be considered in educational domain with respect to models of knowledge to be provided by educational programs.
While software and knowledge architectures have proved similarities [18], in 2HMD approach particular differences between software development and knowledge provision domains are to be considered. Clear analogy between software architecture and knowledge architecture allows assuming, that several aspects of MDA and 2HMD approach can be used for knowledge modelling in the task of study program development [9].

Two-hemisphere metaphor can be transferred to architectural knowledge development. Figure 3 makes a direct projection of software architecture elements into knowledge architecture elements in the context of study module integration into an existing study program.

![Figure 3: Model transformation from problem domain level of knowledge representation into implementation domain level according to 2HMD approach](image)

When we have a task of a new study module integration into an existing study program, content of existing study program and a knowledge field of an integrated module is considered as a problem domain model, where the taxonomy of concepts of new knowledge area forms are a part of conceptual model, which can be enriched by knowledge reflected in the process model, which, in turn, is constructed on the basis of taxonomy elements descriptions given in study program.

Application domain refers to knowledge provision where sub-process model is derived from process model by selecting all processes of exact course of study program, which have been modified accordingly to a new structure of conceptual model and possible interactions between courses defined on the sub-process model and conceptual model.

Due to the difference of problem domains, the task of knowledge modelling for knowledge architecture development can be more straight-forward than software architecture development [9], [11]. In case of knowledge architecture all knowledge reflected in problem domain model has to be presented also at the application domain level instead of looking for processes to be automated as it is in the task of software development. As regards knowledge architecture there is no need to look for process to be automated because all the processes have the same status. The selection of processes for solution level is defined based on two factors, process has to be performed by exact course and the information flows of this process have to be defined according to the concept of integrated knowledge area. Therefore 2HMD approach [1] may be applied to meet knowledge provision needs more closely.

In the case of study module integration into existing study program processes of exact performers (courses of study program) are selected for detailed investigation. All the transformations defined by 2HMD approach for software component development can be applied for a new type of problem domain knowledge, because there is an analogy between the task of analysis of exchange of knowledge flows in studying a particular course and its responsibilities encapsulation and the software class definition based on main benefits of object-oriented software development. Therefore structure of software classes in the interpretation of it in terms of study program elements can be the following:
- class = knowledge unit, which encapsulates an internal structure (attributes) and behaviour (methods)
- attribute = terms or artefacts for studying
- methods = its realization process for practical skills
- generalization (is-a) relationship between classes = suitable for knowledge unit specification
- aggregation (part-of) relationships between classes = suitable for knowledge unit unification

Figure 3 shows 2HMD scheme and information flows from one level to another according to new context of architecture development and adaptation of 2HMD approach for the task of knowledge modelling.

4. INTEGRATION OF KNOWLEDGE FIELD “MODEL DRIVEN ARCHITECTURE” INTO STUDY PROGRAM “COMPUTER SYSTEMS”

To demonstrate described facilities of 2HMD approach the defined transformations are applied to integrate a new field in software systems development, which can be Principles of Model Driven Architecture by itself, into one of the study programs, which are implemented in the Faculty of Computer Science and Information Technology, namely “Computer Systems”, that provides bachelor, master and doctoral level studies. Taking into account teaching resources available for implementation of the new study module, as the first iteration it is being tried to integrate defined knowledge field into the course network which is being carried out by one of the departments of the Faculty of Computer Science and Information Technology, namely, department of Applied Computer Science. Allocation of department’s courses for “Computer Systems” study program is the following:

- Bachelor study program
  - B1 Introduction to Object-Oriented technology
  - B2 Object-Oriented Programming
  - B3 Computer Organization and Assemblers
  - B4 Evolution of Software Technologies
  - B5 XML Technology in Electronic Documents
  - B6 Introduction to Applied Computer Science
  - BD Bachelor’s thesis

- Academic master study program
  - AM1 Object-oriented System Analysis
  - AM2 Software Quality
  - AM3 Applied Computer Science Seminar
  - AM4 Process programming
  - AM5 Tools for Object-oriented System Development
  - AM6 Object-oriented Software Evolution
  - AM7 Methods and Development Trends of Applied Computer Science AMD Master's thesis

- Professional master study program
  - PM1 Object-Oriented Technology and Tools for Large Software Systems Development
  - PM2 Concurrent Processes and ADA Language
  - PM3 Object-Oriented Programming (practical work)
  - PM4 Applied Computer Graphics
  - PM5 HTML
  - PM6 Visual Programming
  - PM7 PROLOG and Logical Programming
  - PM8 Modern Programming Languages (practical work)
  - PM9 Distributed Data Processing in Computer Networks
  - PM10 Technical Cybernetics
  - PM11 Software Risk Analysis
  - PMD Master's thesis

- Doctoral study program
  - D1 Conceptual Aspects of Object-Oriented Programming
  - D2 Conceptual Aspects of Model-driven Software Development
  - D3 Embedded System Software
  - D4 Review and Future Perspectives of Topological Modelling
  - D5 Trends of Object-oriented Technology Evolution
  - D6 Visual Programming Methodologies
This network of courses is a base of left hemisphere in 2HMD approach applied to the task of study module integration into existing study program. Study program process model is being designed by analyzing current study network structure and annotations of courses [19]. For conceptual model elements (the right hemisphere) the data structures from the new knowledge area should be defined considering interrelations between both hemispheres required for 2HMD approach.

Due to publication format limitations and impossibility to present all the models in the framework of one page, process model of study program [19] and conceptual model of Model Driven Architecture is not included in the paper. But the result of 2HMD approach application for the task of integration of MDA knowledge area into study program “Computer systems” of Riga Technical University at the very high abstraction level is presented in the Table. The table shows encapsulation of particular knowledge objects in the form of a map between study courses and studying components for the particular course, which is preferable for presentation of course content in study program definition.

<table>
<thead>
<tr>
<th>TABLE. Mapping of knowledge area “Model Driven Architecture” into courses of department of Applied Computer Science in realization of study program “Computer Systems”</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Technological Fundamentals</td>
</tr>
<tr>
<td>1.1. History of System Modelling Evolution</td>
</tr>
<tr>
<td>1.2. An Introduction to MDA</td>
</tr>
<tr>
<td>1.3. MDA life cycle</td>
</tr>
<tr>
<td>1.4. MDA models</td>
</tr>
<tr>
<td>1.5. General Conceptions</td>
</tr>
<tr>
<td>2. Formalization tools</td>
</tr>
<tr>
<td>2.1. Development of Models</td>
</tr>
<tr>
<td>2.2. Model Transformations</td>
</tr>
<tr>
<td>3. Technological phases</td>
</tr>
<tr>
<td>3.1. Modelling of Problem Domain</td>
</tr>
<tr>
<td>3.2. Modelling of Application Environment</td>
</tr>
<tr>
<td>3.3. Application Architecture</td>
</tr>
<tr>
<td>3.4. Code Generation</td>
</tr>
<tr>
<td>3.5. Testing</td>
</tr>
<tr>
<td>3.6. Maintenance</td>
</tr>
<tr>
<td>3.7. Project Management</td>
</tr>
<tr>
<td>3.8. Quality Management</td>
</tr>
<tr>
<td>3.9. Phases Implementation in Methodologies</td>
</tr>
<tr>
<td>4. Development Tools</td>
</tr>
<tr>
<td>4.1. UML (Unified Modelling Language)</td>
</tr>
<tr>
<td>4.2. XMI (XML Metadata Interchange)</td>
</tr>
<tr>
<td>4.3. xUML / xTUML (eXtended UML)</td>
</tr>
<tr>
<td>4.4. OCL (Object Constraints Language)</td>
</tr>
<tr>
<td>4.5. ASL (Action Semantics Language)</td>
</tr>
<tr>
<td>4.6. QVT (Query / View / Transformation)</td>
</tr>
<tr>
<td>5. Target Platforms</td>
</tr>
<tr>
<td>5.1. Web Services Framework</td>
</tr>
<tr>
<td>5.2. Platforms</td>
</tr>
<tr>
<td>5.3. Specific Applications</td>
</tr>
<tr>
<td>6.1. Common Warehouse MetaModel (CWM)</td>
</tr>
<tr>
<td>6.2. Domain Specific Language (DSL)</td>
</tr>
<tr>
<td>6.3. EAI</td>
</tr>
<tr>
<td>6.4. EDOC</td>
</tr>
<tr>
<td>6.5. Domain Specific Technologies</td>
</tr>
<tr>
<td>7. Supporting Tools</td>
</tr>
<tr>
<td>7.1. Basics of MDA Tools Construction</td>
</tr>
<tr>
<td>7.2. Eclipse Modelling Framework (EMF)</td>
</tr>
<tr>
<td>7.3. Diagramming and Code Generation tools</td>
</tr>
<tr>
<td>7.4. Model Transformation Tools</td>
</tr>
</tbody>
</table>

By analyzing the mappings of model driven development technological components it was found that some of the courses cannot implement all the needed components. Therefore, additional columns are added for
components 3.1, 3.4, 6.1, 6.2, for which the desired learning level is defined, but there is no certain study courses defined (study courses AMX, AMY, PMX, PMY are added).

5. CONCLUSION

Theoretical and practical experiments with 2HMD approach show (1) that analogy between knowledge to be provided by a particular course and a service to be provided by a software program is useful for developing knowledge modelling approaches and (2) that the knowledge domain can be modelled by methods based on MDA principles [11]. Knowledge modelling and architecture development at the high level of abstraction can be performed by using formal methodologies applied in software development, because in both areas the term of architecture is defined in a similar way – it is a set of components with their own structure, relationships to other components and interfaces for cooperation. 2HMD approach can be successfully applied both in software development and knowledge architecture design. Designed architecture can be represented with a study courses networking map. It is a handy tool for a study program development in the desired knowledge area taking into account existing resources.

The paper offers a description of the program for bachelor, master and doctoral students, which allows integrating knowledge about MDA into the study course. The output of this research is a study of courses network together with methodological materials for module execution and enhancement. The results of this paper can be successfully applied by the universities, which include computer science and information technologies programs. It allows tracking rapid changes in the software development technologies field and satisfying needs of the software engineering industry.

Acknowledgement

The research reflected in the paper is supported by the ESF project „Development of study module for model driven software development within the study program “Computer Systems” (Contract Nr. 2007/0080/VPD1/ESF/PIAA/06/APK/3.2.3.2/0008/0007) and by the European Social Fund within the National Programme "Support for the carrying out doctoral study program's and post-doctoral researches".

References


[12] Bertalanffy von L. Perspectives on General System Theory, Published by George Braziller, 1976


