Several Facilities of Class Diagram Generation from Two-Hemisphere Model in the Framework of MDA

Oksana Nikiforova
Riga Technical University.
Riga, Latvia
oksana.nikiforova@rtu.lv

Natalja Pavlova
Riga Technical University.
Riga, Latvia
natalja.pavlova@rtu.lv

Jurijs Grigorjevs
TietoEnator Ltd
Riga, Latvia
jurijs.grigorjevs@tietoenator.lv

Abstract
In Object Management Group Model Driven Architecture models are the primary artifacts during software development, which are presented at the different levels of abstraction from programming details. Class diagram is the most often used model for visual representation of static aspects of software system, but still the formal generation of class diagram from problem domain is under investigation. The paper describes principles of development of class diagram at the conceptual level by using problem domain knowledge presented in the form of two-hemisphere model. Problem domain is presented as two interrelated models of the most important aspects of a system, namely, business process and concept models. Abilities of class diagram generation from two-hemisphere model are presented as a collection of graph transformations and are demonstrated by practical example for abstract problem domain.

Keywords
Model driven architecture, platform independent model, model transformations, two-hemisphere model, UML class diagram

INTRODUCTION

One of the latest researches goals in software engineering is to find a software development process, which would provide fast and qualitative 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.

MDA introduces an approach to system specification that separates the views on three different levels 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). These models are primary artifacts in software developments process in Model Driven Architecture (MDA) of Object Management Group (OMG) and all the activities are focused on transformation from PIM to PSM and from PSM to code. The very important factor is the quality of PIM, i.e. its capability to adequately represent system under development.

The central element of PIM is the presentation of system structure, which would be independent from further implementation and usually is presented in the form of class diagram in Unified Modeling Language (UML) notation [1], as well as adequate presentation of system dynamic, for which different modeling tools are used. This paper discusses the development of class diagram, which satisfies the main statement of MDA and is based on transformation from two-hemisphere model into elements of class diagram defined in UML notation.

Section 2 defines the transformation of models and precise what is necessary for successful transition from one model to another. The main statement of MDA is that all transformations of the model have to be defined in a precise and comprehensive way. Transformations require transformation definitions between any model, which serve as a source, and the same for the target model. Section 3 describes two-hemisphere model as a source model of transformation into class diagram and defines main steps of two-hemisphere model driven approach suitable for platform independent modeling. Section 4 defines all possible transformations from elements of two-hemisphere model into elements of class diagram, which is defined as a target. The tool developed for approbation of defined transformations as well as practical example of the application of the approach discussed in the paper are briefly demonstrated in Section 5. Section 6 concludes the paper and discusses possible future work.

BASICS ON TRANSFORMATIONS IN THE FRAMEWORK OF MDA

The main idea of MDA is to achieve formal system representation at the highest level of abstraction. Nowadays MDA tools support translation of platform independent system presentation into software components and code generation and researchers try to "raise" it as high as possible to fulfill the main statement of the MDA [2]. One of the most important and problematic stages in MDA realization is derivation of PIM elements from a problem domain and PIM construction in the form that is suitable for the PSM. It is necessary to find the way to develop PIM using formal representation, so far keeping the level of abstraction high enough. PIM model should represent system static and dynamic aspects. Class diagram shows static structure of the developed system. But UML is a modeling language and does not have all the possibilities to specify context and the way of modeling, which is always required to be defined in a methodology. Therefore, the construction of class diagram has to be based on well defined rules for its elements

978-1-4244-2881-6/08/$25.00 ©2008 IEEE

Authorized licensed use limited to: UNIVERSIDAD DE GRANADA. Downloaded on January 20,2024 at 21:00:10 UTC from IEEE Xplore. Restrictions apply.

generation from the problem domain model presented in the suitable form.

Class diagram discussed in the paper contains classes, relations among them, attributes and operations of classes. Dynamic aspects, which are another meaningful component of system presentation at the platform independent level is not the object of the current research. To obtain the class diagram the initial business knowledge represented with two-hemisphere model may be used. The transformation of this model into class diagram is discussed in the paper. The transformation should be defined in formal way and should be acceptable for use in transformation tool. The structure of a transformation tool is discussed in [2] with definition of models, necessary for transformation and transition between these models. Transformation tools take a source model as an input, and create another model, called target model, as an output [2]. Therefore, implementation of transformation needs well-defined set of notational elements of source and target models and definition for transformation of elements of one model into elements of another one.

The paper describes class diagram development based on two-hemisphere model. Therefore, according to Kleppe's definition source model is defined in terms of two-hemisphere model (business process and concept model) and target model is defined in terms of UML class diagram.

DESCRIPTION OF TWO-HEMISPHERE MODEL

Two-hemisphere model driven (2HMD) approach [3] proposes using of business process modeling and concept modeling to represent systems in the platform independent manner and describes how to transform business process models into UML models. For the first time the strategy was proposed in [4], where the general framework for object-oriented software development had been presented and the idea about usage of two interrelated models for software system development has been stated and discussed. The title of the proposed strategy [3] is derived from cognitive psychology [5]. Human brain consists of two hemispheres: one is responsible for logic and another one for concepts. Harmonic interrelated functioning of both hemispheres is a precondition of an adequate human behavior.

Essence of Two-Hemisphere Model

A metaphor of two hemispheres may be applied to software development process because this process is based on investigation of two fundamental things: business and application domain logic (processes) and business and application domain concepts and relations between them. Two-hemisphere approach proposes to start process of software development based on two-hemisphere problem domain model, where one model reflects functional (procedural) aspects of the business and software system, and another model reflects corresponding concept structures. The co-existence and inter-relatedness of these models enables use of knowledge transfer from one model to another, as well as utilization of particular knowledge completeness and consistency checks [3]. Figure 1 shows the schema of 2HMD approach for system development.

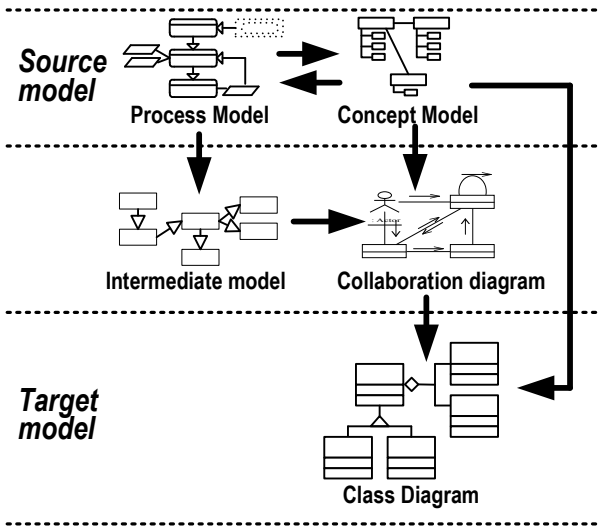

Figure 1. Transformations from two-hemisphere model into class diagram under 2HMD approach.

A notation of the business process model is optional, however, it must reflect the following components of business process model: processes, performers, information flows, and information (data) stores [3]. Real-world classes relevant to the problem domain and their relationships are presented in concept model. It is a variation of well known entity relationship (ER) diagram notation [6] and consists of concepts (i.e. entities or objects) and their attributes. The notational conventions of the business process diagram give a possibility to address concepts in concept model to information flows (e.g. events) in process model.

Some recent surveys show that about 83% of companies are engaged in business process improvement and redesign. This implies that many companies are common with business process modeling techniques or at least they employ particular business process description frameworks. On the other hand practice of software development shows that functional requirements can be derived from problem domain task descriptions even about 7 times faster than if trying to elicit them directly from users. Both facts mentioned above and existence of many commercial business modeling tools are a strong motivation to base software development on the business process model rather than on any other soft or hard models. The investigation of 2HMD approach under the MDA framework shows that approach could be applied for generation of several elements of class diagram, namely, in addition to generate class names, attributes and several methods [7]. This paper shows the strategy of two-hemisphere model application for generation of UML class diagram.

Transformations from Two-Hemisphere Model into Elements of Class Diagram

Discussed way of application of 2HMD approach gives a possibility to define different types of relationships be-

Authorized licensed use limited to: UNIVERSIDAD DE GRANADA. Downloaded on January 20,2024 at 21:00:10 UTC from IEEE Xplore. Restrictions apply.

tween classes and to share method responsibilities in a more precise way. The detailed transformations between models are shown in Figure 2.

Two-hemisphere model consists of business process model (graph G1 on Figure 2) and concept model (graph G2 on Figure 2).The notation of business process model does not have a significant value, main requirement to the notation of business process model is possibility to define business processes, performers, events and data flows among business processes. For current research business process model constructed with GRAPES [8] notation is used. The second hemisphere is concept model. C.Larman defines concept model as: "The concept model captures real-world concepts (objects), their attributes, and the associations between these concepts." [9].

To perform transformation to class diagram the intermediate model (graph G3 on Figure 2) is introduced. Intermediate model is used to simplify the transition between business process and object interaction models, which is now presented in the form of UML collaboration diagram (graph G4 in Figure 2). Figure 2 shows all the transformations from the business process model (G1) and concept model (G2) into the class diagram (G5). Transformations are based on the hypothesis that elements of the class diagram can be received from the two-hemisphere model by applying defined techniques of graph transformation [10].

Intermediate model is generated from business process model using methods of directed graph transformation, when arcs of one graph (G1 on Figure 2) are transformed into nodes of another graph (G3 on Figure 2) and nodes of one graph (G1) are transformed into arcs of another graph (G2) [7]. Figure 2 presents the sequence of transformations from two-hemisphere model into class diagram with dotted arrows. Business process "perform action 1" is transformed into arc "perform action 1" of intermediate model (graph G3 on Figure 2). The next transformation creates the method "perform action 1()" in collaboration diagram (graph G4 on Figure 2) from the arc of intermediate model. The last transformation of this business process defines the responsible class of this method in class diagram (graph G5). The element "performer 1" is transformed as a node of intermediate model, and as "actor 1" of collaboration model. This element is defined as "actor 1" in class diagram. Data types for elements "event 1" and "event 3" are defined as "DataType A" or "Concept A" of concept model.

Events are transformed into nodes of intermediate model, and then into objects like "Event1: Class A" in collaboration diagram, which serves as a base for classes of class diagram definition. All attributes for classes are determined based on attributes defined in concept model.

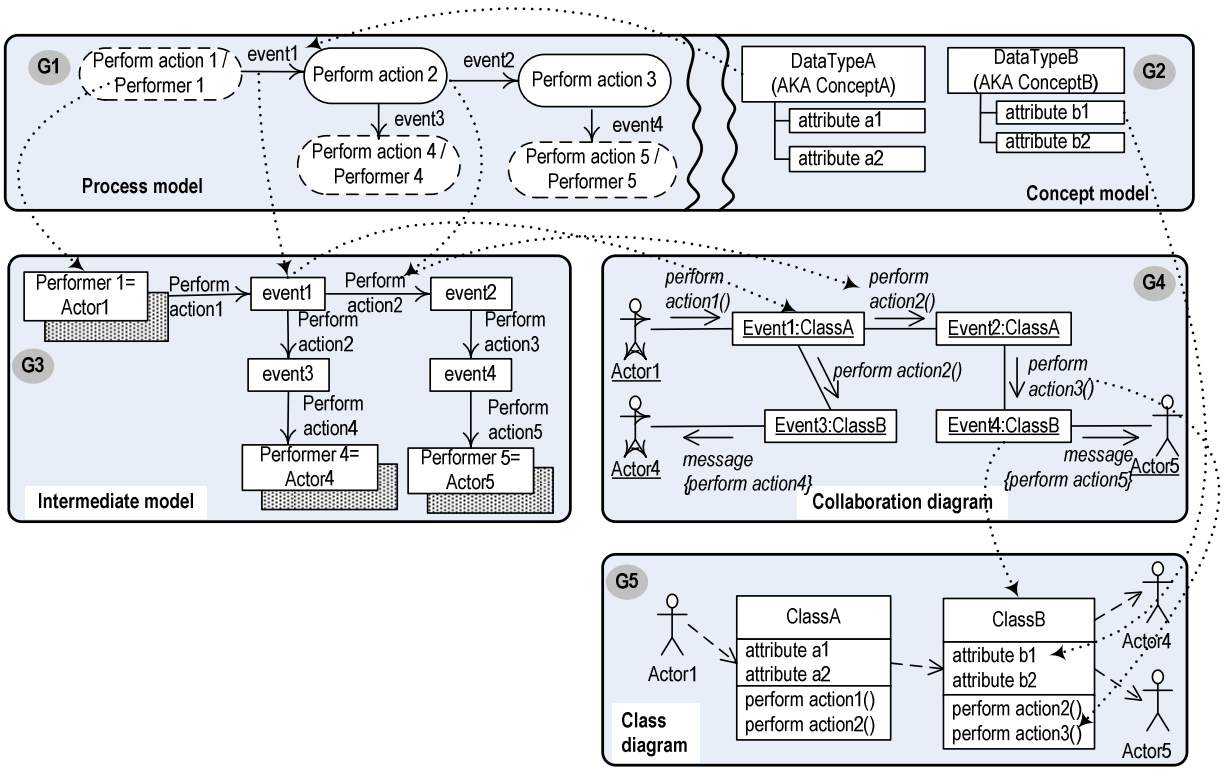

Figure 2. Essence of application of two-hemisphere model for generation of elements of class diagram.

Authorized licensed use limited to: UNIVERSIDAD DE GRANADA. Downloaded on January 20,2024 at 21:00:10 UTC from IEEE Xplore. Restrictions apply.

Simple Example of Application of Two-Hemisphere Model for Generation of Elements of Class Diagram

For better understanding of the main idea, the example of such model transformations is shown for a fragment of problem domain concerned with room reservation in the hotel [9]. Figure 3 presents only fragment of transformation. There is one process "Receive reservation blank", which has output "empty reservation blank". Concept "reservation blank" defines data type for the output of process. It is transformed into fragment of intermediate model with arc "Receive reservation blank" and node "Empty reservation blank". Intermediate model allows to receive collaboration diagram, where initial process "Receive reservation blank" is a method of object "Empty reservation blank".

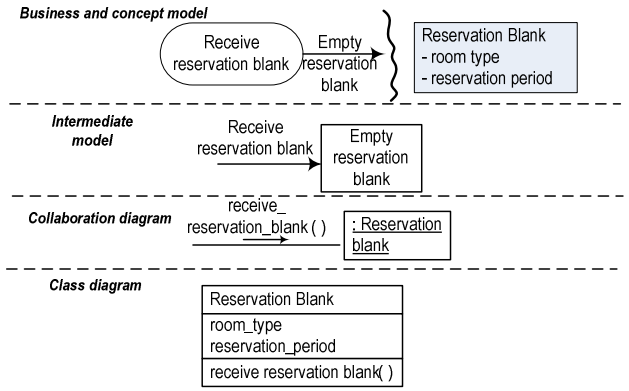

Figure 3. An example of transformation of process and concept elements into class elements.

Object "Empty reservation blank" belongs to class "Reservation blank", which is defined with corresponding concept. When a collaboration of objects is defined, it is possible to construct class diagram according to rules of object-oriented system modeling [4].

SET OF TRANSFORMATIONAL ABILITIES OF TWO-HEMISPHERE MODEL INTO CLASS DIAGRAM

During the investigation of receiving of class diagram from two-hemisphere model all possible combinations of number and types of incoming and outgoing information flows from nodes of processes are examined [9]. Different combinations give a possibility to receive different relations among classes. The paper discusses transformations, which help to define aggregation, dependency and inheritance, as more interesting cases, which demonstrate advanced facilities of application of two-hemisphere model.

Transformation of processes with single input and single output has two possible combinations, but only one of them gives possibility to receive aggregated classes (see Figure 4). Data types are different in this case, respectively, data types for flows "Room (Found room number) and "Room status" are defined by concerned concepts "Room" and "Room Status". This combination is demonstrated with real example presented for problem domain of hotel room reservations. Figure 4 displays how from process and concept model fragment the aggregated classes "Room" and "Room status" could be created. The intermediate model, received from process model serves as a base for collaboration diagram. With collaboration diagram two objects and method are defined, which is executed from object "Room status" with object "Room". The information of concept, process and collaboration diagram are enough for definition of class diagram fragment with two classes, aggregation between them, attributes and method in one of them.

Transformation of processes with multiple inputs and single output has more different combinations. One of it gives a possibility to define dependency. It is demonstrated in Figure 5. Case shown on Figure 5 contains combinations of different types of incoming and outgoing data flows.

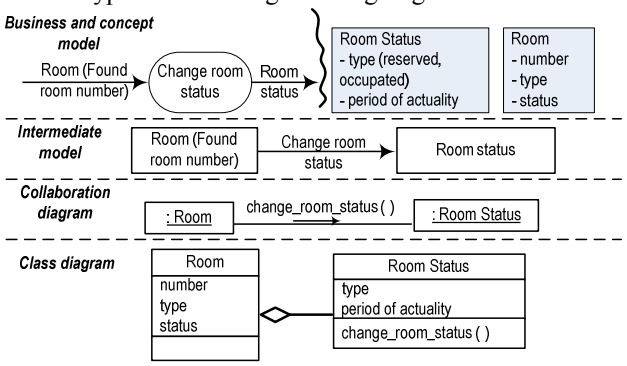

Figure 4. Definition of aggregation relationship.

Process "Give reservation information to client" with two inputs and one output is presented in Figure 5. Data flows "Room Status" and "Reservation Information" are defined with two corresponding concepts. Process model is transformed into intermediate model in two steps (see Figure 5). The first one is simple transformation from intermediate model. The second one joins two objects "Reservation Information" into one, for further transformation into class diagram. As the result the class diagram has two classes, relations "dependency" between them, attributes and one method.

Transformation case, when process has one input and two outputs with the same type give a possibility to define generalization. This case is shown in Figure 6. The process "Check room status" has data flow "Room (room number) as input and two data flows "Free room" and "Reserved room" as output. For all these data flows data type is defined with one concept "Room". Transformation into collaboration diagram is performed with two steps: the first one has three similar objects and one message among them, and the second one has one object "Room", joined from three objects from previous step. The class diagram has three classes, where class "Room" generalize two sub-classes "Free room" and "Reserved room". Thus, three different types of class relationships are discussed in this section.

Authorized licensed use limited to: UNIVERSIDAD DE GRANADA. Downloaded on January 20,2024 at 21:00:10 UTC from IEEE Xplore. Restrictions apply.

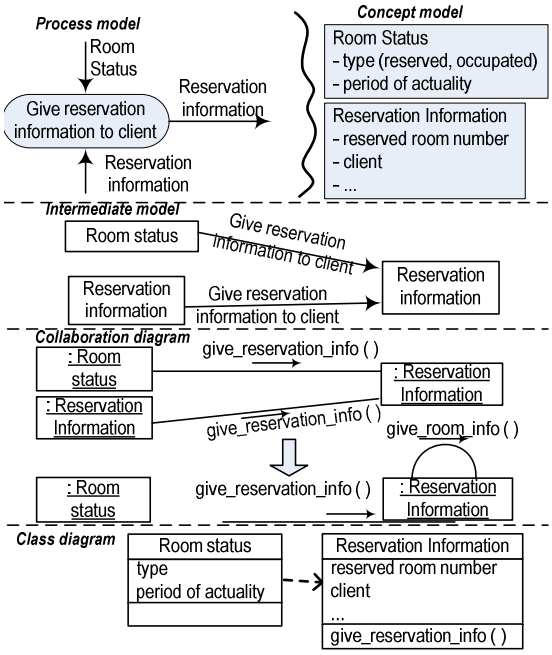

Figure 5. Definition of dependency relationship.

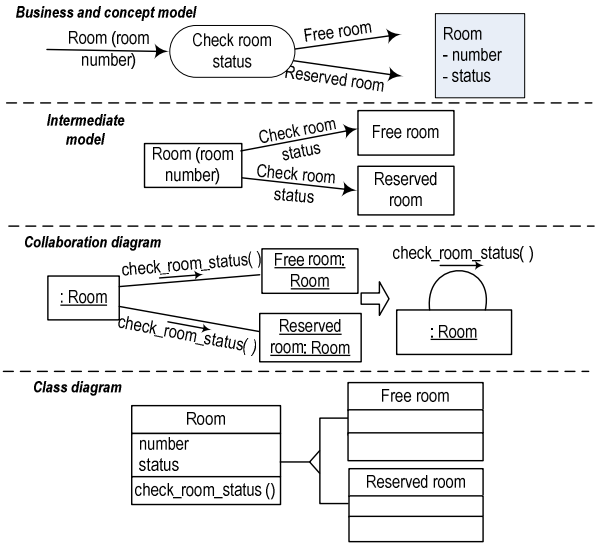

Figure 6. Definition of inheritance relationship.

AUTOMATION ABILITIES OF TRANSFORMATIONS FROM TWO-HEMISPHERE MODEL INTO CLASS DIAGRAM

The transformation flow described above allows to define the requirements and algorithm for class diagram generation. It is possible to automate the transformations discussed in the paper. According to the approach demonstrated in the paper the tool for generation of class diagram includes the steps shown in Figure 7: construction of two-hemisphere model; generation of model elements and its interrelations in a structured form; application of the transformation rules; receiving class specification in well structured form suitable for class diagram construction (for example XML format).

The tool for business process modeling GRADE [8] gives a possibility to construct two interrelated models (business process and the concept ones) and to generate text description of models with permanent structure, therefore it is chosen as a tool for development of two-hemisphere model and further generation of textual files, which defines all the elements of the model and their relations each to other. Generated text files serve as input information into the tool developed for support of the processing algorithm and XML file, which contains description of structure of the required class diagram.

XML format of class specification gives a possibility to receive visual representation of class diagram in any tool, which supports import from XML for class diagram development. The tool is applied for generation of class diagram from two-hemisphere model developed for hotel room reservation problem domain and the class diagram shown in Figure 8.

Using the proposed approach, classes, attributes, operations and relations among classes, which could be determined from the business process diagram, and the data structure, were defined. One of the limitations of the approach is an impossibility to define the full specification of methods with its arguments. This could be one of the potential directions for future investigation of application of two-hemisphere model for generation of class diagram elements.

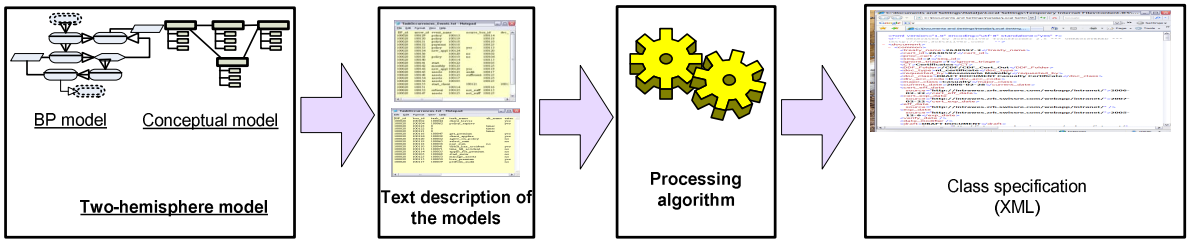

Figure 7. General schema of generation of class specification by tool, which support the transformations defined.

Authorized licensed use limited to: UNIVERSIDAD DE GRANADA. Downloaded on January 20,2024 at 21:00:10 UTC from IEEE Xplore. Restrictions apply.

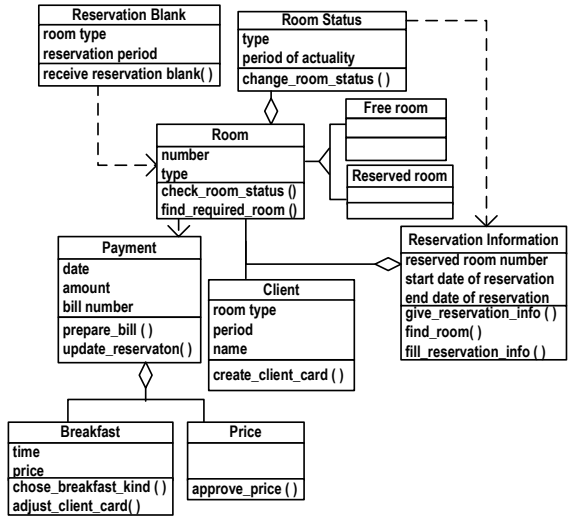

Figure 9. Class diagram for hotel room reservation.

CONCLUSIONS

MDA consists of four models – CIM, PIM, PSM and code. There are information transformations that exist between the above models in order to derive each model from its predecessor-model. Nearly all transitions between the models were already characterized in various researches with one significant exception – there is still no sure that the PIM, which is received from the CIM, corresponds to the PIM, which can be used in the PSM generation. The paper discusses abilities on usage of problem domain knowledge presentation in terms of two-hemisphere model, which contains two interrelated models of system aspects – process and concept presentations. On one hand it gives a possibility to present knowledge in a form understandable for both business user and system analyst, moreover cover complete and consistent presentation of different system aspects, and on the another hand it supports the formal transformations of model elements into elements of UML class diagram, which often is a starting point during software development. The central hypothesis of this research is that it is possible to apply the graph theory technique for model transformation in the framework of MDA, where the source model is defined in terms of a business process model, associated with a concept model, and the target model is defined in terms of a class diagram. Analysis of the system models presented as a set of graphs developed on the basis of the initial two-hemisphere model enables derivation of the class diagram, which is the central component of PIM and is sufficiently detailed in order for the PSM to be generated. The proposed transformations are applied for two-hemisphere model of hotel room reservation and classes with attributes and different kinds of relationships are identified based on elements of process and concept models. The ability to define all the types of transformations in a formal way gives a possibility to automate the process of class diagram development from correct and precise two-hemisphere model.

REFERENCES

[1] OMG UML, Unified Modeling Language Specification / Retrieved from: http://www.omg.org.

[2] Kleppe A., Warmer J., Bast W., 2003. MDA Explained: The Model Driven Architecture – Practise and Promise, Addison Wesley, 192 p.

[3] Nikiforova O., Kirikova M., 2004. Two-Hemisphere Model Driven Approach: Engineering Based Software Development, Proceeding of the 16th International Conference Advanced Information Systems Engineering Caise'2004. A. Persson And J. Stirna (Eds.), Lncs 3084, Springer – Verlag Berlin Heidelberg, pp. 219 – 233

[4] Nikiforova O., 2002. General Framework For Object-Oriented Software Development Process. Proceedings of Conference of Riga Technical University, Computer Science, Applied Computer Systems, 3rd Thematic Issue, Pp.132-144, Riga, Latvia

[5] Anderson, J.R., 1995. Cognitive Psychology and Its Implications, W.H. Freeman and Company, New York.

[6] Chen P., 1976. The entity relationship model – towards a unified view of data. ACM Trans. Database Systems, 1, 9-36

[7] Nikiforova O., Kirikova M., Pavlova N., 2007 Principles of Model Driven Architecture in Knowledge Modeling for the Task of Study Program Evaluation, Databases and Information Systems IV, by IOS Press in the series "Frontiers in Artificial Intelligence and Applications", pp.291-304

[8] GRADE tools, GRADE Development Group// Internet. -*http://www.gradetools.com/*

[9] Pavlova N., 2008. Approach for Development of Platform Independent Model in the Framework of Model Driven Architecture, Ph.D. thesis, Riga Technical University

[10] Grundspenkis J., 1997 Causal Domain Model Driven Knowledge Acquisition for Expert Diagnosis System Development // Kaunas: Kaunas University of Technology Press.

Authorized licensed use limited to: UNIVERSIDAD DE GRANADA. Downloaded on January 20,2024 at 21:00:10 UTC from IEEE Xplore. Restrictions apply.