# Development of the Tool for Generation of UML Class Diagram from Two-Hemisphere Model

Oksana Nikiforova, Natalya Pavlova
*Department of Applied Computer Science, Riga Technical University, Riga, Latvia*
*oksana.nikiforova@rtu.lv, natalja.pavlova@rtu.lv*

## Abstract

*Nowadays, the research on the Model Driven Architecture is still in progress. One of the existing problems in this area is transformation between models of the Model Driven Architecture. This paper discusses the transformation from Platform Independent Model, received from Computation Independent Model, to the one, ready for generation of Platform Specific Model. To implement this transformation, two-hemisphere model driven approach to system modeling is chosen. The class diagram in UML notation is defined as transformation result. The paper discusses the transformation into class diagram, and the possibility of automation of such transformation. In order to research automation possibilities, custom tool is designed and implemented. This tool allows generating elements of class diagram based on provided transformations. It is applied on three examples of initial model, represented in different notations and different problem areas. These experiments proof that examined transformations are independent from problem domain and notation which expresses initial knowledge.*

## 1. Introduction

Since beginning of eighties a numerous accounts of model generated software systems have been offered to attack problems regarding software productivity and quality [1]. CASE tools developed up that time were oversold on their “complete code-generation capabilities” [2]. Nowadays, similar arguments are exposed to Object Management Group (OMG) Model Driven Architecture (MDA) [3], using and integrating Unified Modeling Language (UML) models [4] at different levels of abstraction. Manipulation with models enables software development automation within CASE tools supported by MDA [5], [6], [7].

One of the actual research goals in the area of software engineering is selection of software development process, which could provide fast and qualitative software development. Most of currently proposed methodologies and approaches tires to make the development process easier and still more qualitative. In order to achieve this goal the role of explicit models becomes more and more important. Lately, the most popular approach is Model Driven Architecture [8].

MDA is the central component in the OMG’s strategy for maximizing return on investment, reducing development complexity and future-proofing against technological change [9]. MDA tools do not support the complete code-generation capabilities from the initial business information, and the most problematic stage is system modeling based on knowledge about problem domain [10].

An idea of MDA is promising – raising up the level of abstraction, on which systems are developed. It could lead to more qualitative development of complex systems. The main idea of MDA is to achieve formal system representation at the highest possible level of abstraction. Development process is defined as a sequence of steps, when four types of models are created: Computation Independent Model (CIM), Platform Independent Model (PIM), Platform Specific Model (PSM), and Program Code. Nowadays MDA tools support translation of solution elements into software components, for example, transformation from PIM to PSM, and code generation. Researchers try to “raise” level of abstraction as high as possible to fulfill the main statement of the MDA [8]. 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 suitable for the PSM. It is necessary to find the way of PIM development using formal representation, so far keeping the level of abstraction high enough.

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 UML notation [11], as well as

978-0-7695-3372-8/08 $25.00 © 2008 IEEE
DOI 10.1109/ICSEA.2008.37

adequate presentation of system dynamics. Different modeling tools are used for that. The paper discusses the class diagram development aspects, which satisfies the main statement of MDA and are based on transformation from two-hemisphere model into elements of class diagram defined in UML.

Currently, transformations between UML models are still a subject of intensive investigation. Principles of simple language for transformations are presented in [6]. Several proposals [12], [13] are made in response to OMG request for proposals to MOF Query/View/Transformation [14]. The great attention is devoted to UML class diagram development, because class diagram in UML-based CASE systems serves as a main source of knowledge for development of software system: database specification, graphical user interface, application code, etc. [15]. The goal of this paper is to propose CASE tool for development of class diagram structure at the conceptual level by using problem domain knowledge presented in the form of two-hemisphere model introduced in [10], [16], [17], which consists of two interrelated models of system aspects – business process and concept presentation.

Section 2 discusses possibilities of class generation and relations among classes based on initial knowledge represented by business process and concept model. Section 3 discusses design of tool intended for generation of classes and their relations. Section 4 presents the results of constructed tool application, as well as the class diagram, which was constructed by the tool. Finally, Section 5 provides research conclusions.

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

Two-hemisphere model driven (2HMD) approach [17] proposes the usage 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. Development of platform independent model approach is illustrated in Figure 1 [18], [19] in the terms of class diagram according to two-hemisphere model driven. For the first time the two-hemisphere model driven strategy is proposed in [16], 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 [10] is derived from cognitive psychology [20]. Human brain consists of two hemispheres one of which is responsible for logic and another one for concepts. Harmonic interrelated functioning of both hemispheres is a precondition of an adequate human behavior.

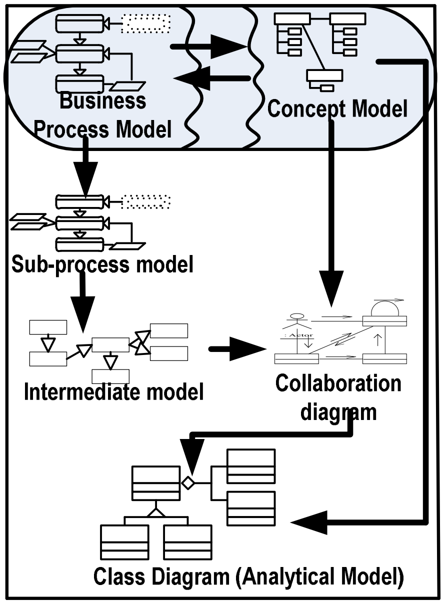

Figure 1. General look on transformations from two-hemisphere model into class diagram

A metaphor of two hemispheres may be applied to software development process, mainly because this process is based on investigation of two fundamental things: business and application domain logic (processes) and business and application domain concepts and their relations. 2HMD approach proposes software development process initiation 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 conceptual structures. The co-existence and inter-relatedness of models enables use of knowledge transfer from one model to another, as well as utilization of particular knowledge completeness and consistency checks [17].

A notation of the business process model (graph G1 in Figure 2), which reflects functional perspectives of the problem and application domains, is optional, however, it must reflect the following components of business processes: processes; performers; information flows; and information (data) stores, see Figure 2 [17].

The concept model (graph G2 in Figure 2) is used in parallel with business process model to cross-examine software developers understanding of problem and platform independent models. Real-world classes relevant to the problem domain and their relationships are presented in concepts model. The notational conventions of the business process diagram gives a possibility to address concepts in concept model to information flows in process model (see Figure 2).

Transformation from two-hemisphere model into elements of class diagram at the abstract level is shown in Figure 2. The schema presents the way how elements of business process model (graph G1 in Figure 2) and concept model (graph G2 in Figure 2) are transformed into elements of class diagram (graph G5 in Figure 2), using intermediate model (graph G3 in Figure 2) and collaboration diagram (graph G4 in Figure 2) [21].

If business process model is constructed at high level of abstraction and has superficial description of business processes, then each process could be described in details with construction of sub-process models, using the same notation as for business process model. If the sub-process models are being constructed, these models supplant the business process model (graph G1 in Figure 2) within the chain of transformations.

Intermediate model (graph G3 in Figure 2) is used to simplify the transition between business process or sub-process and object interaction models, which is presented in the form of UML collaboration diagram model (graph G4 in Figure 2).

Intermediate model (graph G3 in Figure 2) is generated from business process or sub-process models using methods of direct graph transformation and is presented as graph with nodes and arcs among nodes.

Transformations are based on the hypothesis that elements of the class diagram (graph G5 in Figure 2) can be received from the two-hemisphere model by applying defined techniques of graph transformation [22]. The nodes of the graph G1 of Figure 2 (business process or sub-process model) become arcs of the graph G3 of Figure 2 (intermediate model), and arcs of the graph G1 of Figure 2 (business process or sub-process model) become nodes of the graph G3 of Figure 2 (intermediate model) [23].

Constructed intermediate model serves as a base for collaboration model. The collaboration diagram is represented as a graph G4 in Figure 2, where arcs are defined data structures and arrows are defined as performed actions.

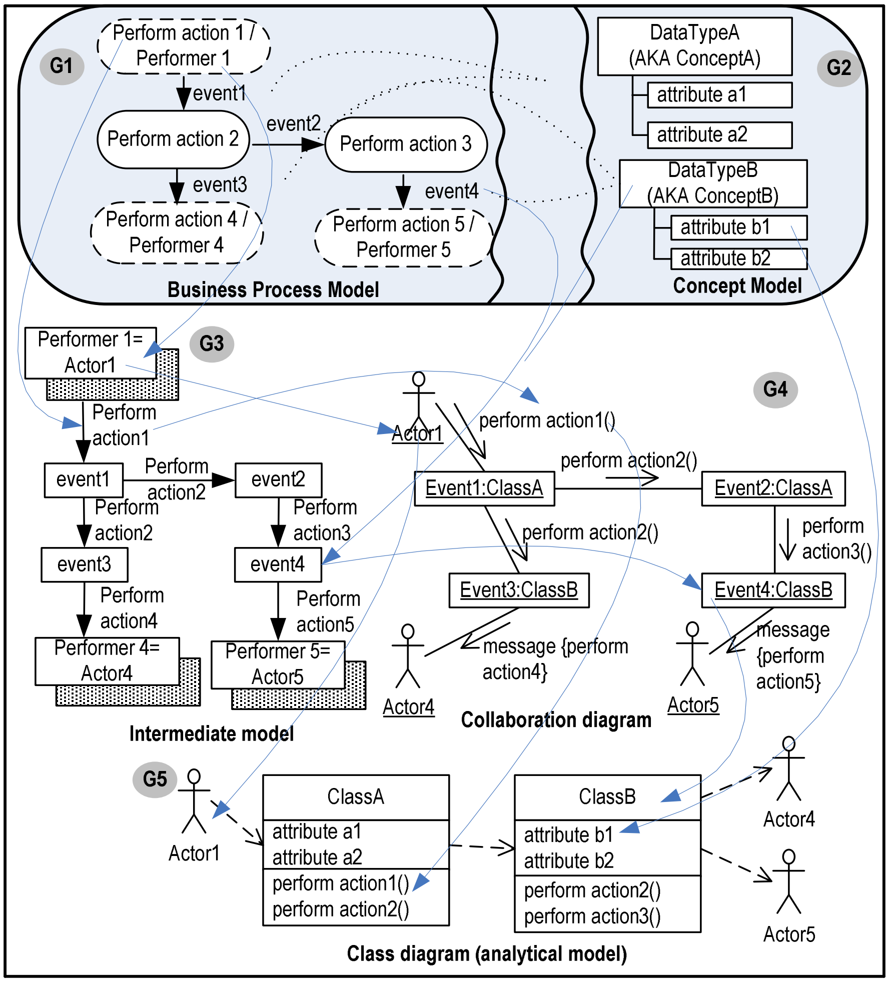

Figure 2. Essence of application of two-hemisphere model for generation of elements of class diagram

Therefore if each process is examined as a message, and each data flow as an object, a draft collaboration diagram could be received by replacing all events of intermediate model with concerned class exemplars and the actions of intermediate model with messages or operations.

The next state of transition is a class diagram. Here all processes of the graph G1 of Figure 2 (business process or sub-process model) are operations of classes, and all events and concepts of graphs G1 and G2 (business process or sub-process model and concept model) are classes or instances of classes. Class diagram presents the set of attributes based on attributes defined in concept model.

In a very simple example, transformation described above looks like it is shown in Figure 3, where transformation of process and concept elements into class elements is represented. 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 systems [23]. An analysis of all possible combinations gives a possibility to define different types of relationships between classes, to share class responsibilities and to encapsulate class attributes and methods. Based on theory of combinatorics [24] it is possible to define different combinations of arcs and nodes in business process or sub-process model, which are structured as the following collection:

- transformations of business processes with single input and single output;
- transformations of business processes with multiple inputs and single output;
- transformations of business processes with single input and multiple outputs;
- transformations of business processes with multiple inputs and multiple outputs.

All possible transformations are described in details in [10], [23]. The paper offers the description of tool for the usage of two-hemisphere model for class diagram generation based on the defined transformations.

## 3. Transformation Tool

The recent tendency of automation of information handling process is essential in industry of information technology [25]. It gives a possibility to spare human and time resources.

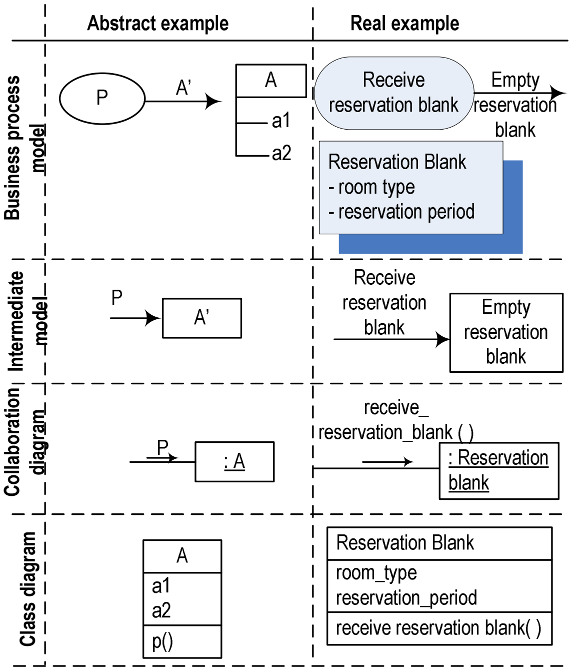

Figure 3. An example of process and concept elements transformation into class elements

The implementation of tool, which automates transformation into class diagram, gives a possibility to receive static structure of the system without spending of a lot of time on design. For any transformation the initial data and needed result should be defined before.

Transformation tools takes a model, so called source model, as an input, and creates another model, so called target model, as an output [6]. General schema of transformation tool is shown in Figure 4. Therefore implementation of model transformation (in our case transformation from two-hemisphere model into class diagram) needs well-defined set of notational elements of source model, well-defined set of notational elements of target model and definition for transformation of elements of one model into elements of another one.

Section 2 outlines development of UML class diagram based on two-hemisphere model, therefore according to schema in Figure 4, source model is defined in terms of two-hemisphere model and target model is defined in terms of UML class diagram.

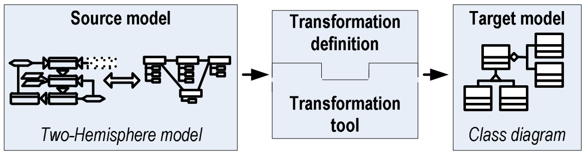

Figure 4. Schema of model transformation tool

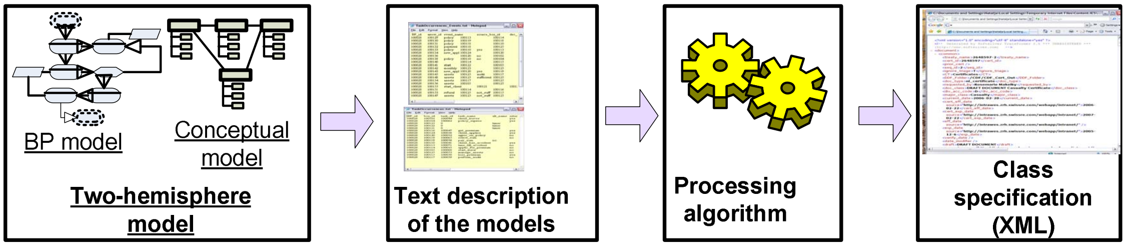

Figure 5. Steps of generation of class specification from two-hemisphere model by the tool described

The transformations discussed in Section 2 and described in [10], [23] allows defining the requirements and algorithm for class diagram generation, which enables transformation automation from two-hemisphere model into class diagram.

Class diagram generation should consist of four steps according to the application of two-hemisphere model (see Figure 5):

- construction/receiving of two-hemisphere model;
- generation of model elements and their interrelations in some structured form;
- application of the transformation rules defined (processing algorithm);
- receiving class specification in well structured form suitable for class diagram construction (for example, XML format).

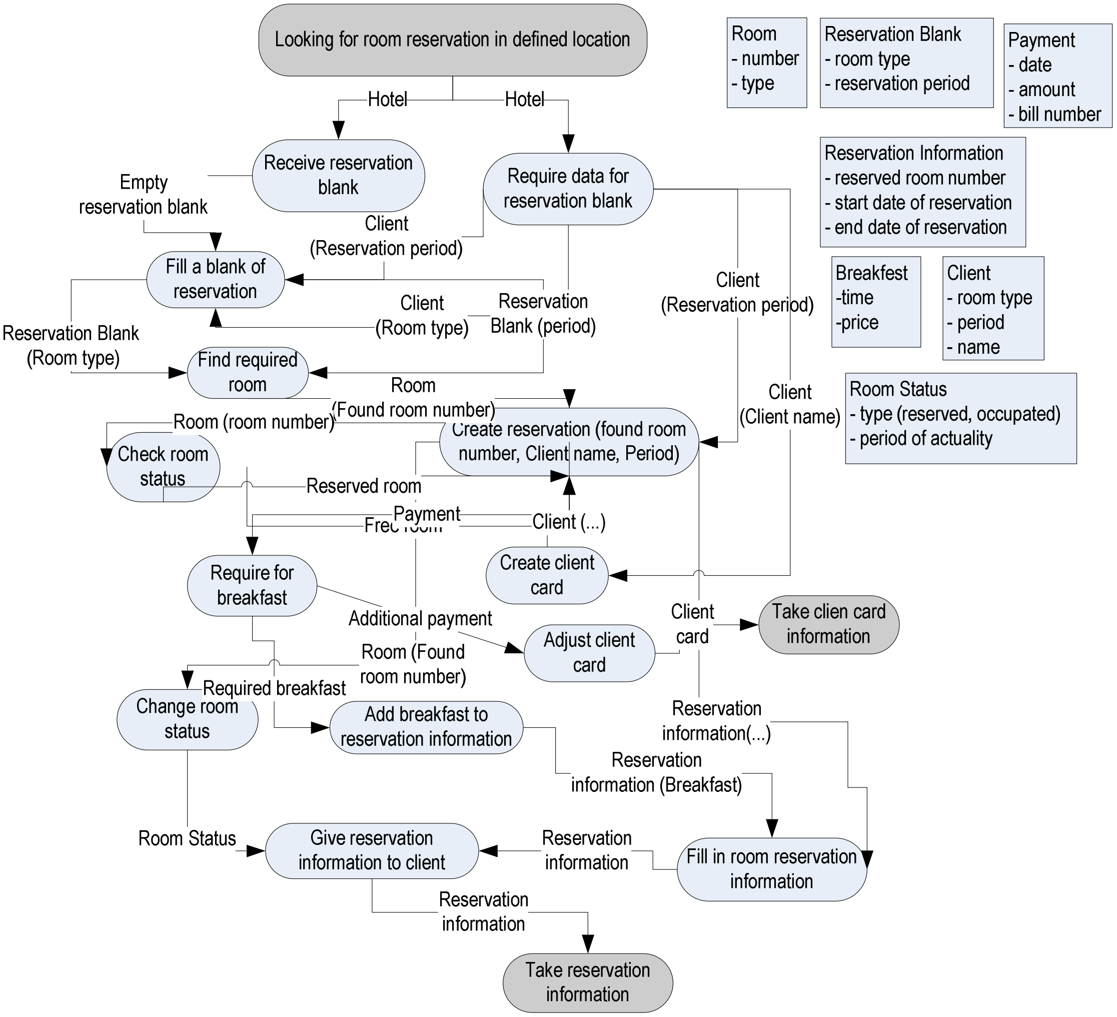

Figure 6. Initial two-hemisphere model of hotel room reservation

When the structures of input and output data are known, it is possible to automate a process of input data transformation into output data. But in this case, the tool for development of two-hemisphere model with the possibility to define model elements and their relations in a structured form is required. One of the tools for business process modeling, which gives a possibility to construct two interrelated models (business process and the concept ones), is GRADE [26]. Indeed, GRADE generates text descriptions of model with permanent structure, therefore it is chosen as a tool for development of two-hemisphere model and further generation of textual files. It defines all the elements of the model and their relations from one into another.

Generated text files serves as an input information into the tool developed in order to support the processing algorithm and XML file, which contains structure of the class diagram required. 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. Existence of automated tools for receiving class structures proves, that discussed transformations are enough formal for programming.

The tool is applied for generation of class diagram from two-hemisphere model developed for hotel room reservation problem domain. Two-hemisphere model of hotel room reservation is shown on Figure 6.

Classes, attributes, operations and relations among classes, which could be determined from the business process diagram and the data structure, were defined applying discussed transformations from business process and concept model to class diagram. Figure 7 represents obtained structure of class diagram for hotel room reservation.

In order to verify correctness and independency from business process and data structure modeling notation, the transformations were applied on different types of examples.

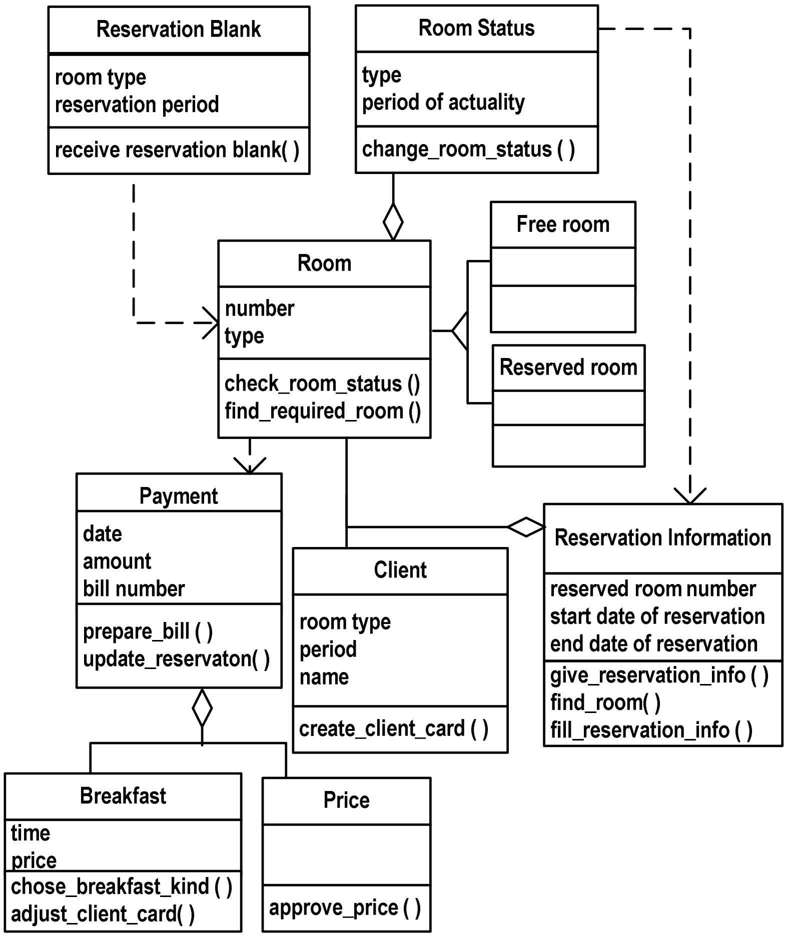

Figure 7. Target model – class diagram for hotel room reservation

## 4. Application of results

To recheck obtained theoretical and practical objectives defined, transformations from two-hemisphere model into class diagram were applied to three examples: (1) problem area of hotel room reservation, where initial business process model is constructed in GRAPES notation [27] with CASE tool GRADE [26]; (2) problem area of hotel room reservation, where initial business process model is constructed in IDEF0 notation [28] with CASE tool BPWin; (3) problem area of insurance system, where initial model is constructed in GRAPES notation with CASE tool GRADE [26] (the difference with the case 1 is that the model was constructed by developers of GRADE tool as demo example, and the different problem domain is chosen).

Business process diagram for IDEF0 notation is constructed in BPWin tool. Unfortunately, it does not provide construction of concept or object model. Therefore attributes of classes, received for hotel room reservation system with initial information in IDEF0 notation are missing in the class diagram. Even in this case automation of distributing methods among classes is important contribution within software development.

For insurance problem domain, business process model is constructed in GRAPES notation using CASE tool GRADE, where initial model of insurance system is built-in example of CASE tool GRADE (demo example in other words). This experiment was performed to proof that transformations described in Section 2 are independent from modeler of problem domain, and to avoid the subjective point of view of modeler during business process and concept model construction. The class diagram, received in this experiment, displays static structure of insurance system.

The results of automated tool activities is the specification of class structure in XML, which could serve as a base of class diagram construction in any tool, which supports UML or MDA.

Experiments on applying discussed transformations from two-hemisphere model into class diagram in different problem domain prove that transformations are independent from problem domain.

Experiments on applying transformation from two-hemisphere model, constructed in various CASE tools and notations, prove that transformations from two-hemisphere model into class diagram are independent from used notation of business process modeling, as well as CASE tool, used for initial model creation.

## 5. Conclusions

MDA consists of four models – the CIM, the PIM, the PSM and the code. There are transformations that exist between the above models in order to derive each model from its predecessor. Nearly all transitions between the models are already characterized in various researches with one significant exception – there is still no confidence that the PIM, which is received from the CIM, corresponds to the PIM, which can be used for 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 presentation.

According to [30] the significant aspect of real world behavior is seen from the process point of view, where process is understood as a collection of actions, chronologically ordered and influencing objects and is more then "just an amorphous heap of the action". Two-hemisphere model corresponds to both fundamental things - functional aspects of the system defined in terms of business processes and the structural ones defined in terms of concept model.

The proposed transformations are applied to two-hemisphere model of hotel room reservations and classes with attributes. 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. On one hand, it enables knowledge representation in a form understandable for both business users 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. Two-hemisphere model gives a possibility to define classes with attributes and operations they have to perform, as well as different types of relations can be defined between classes, based on analysis of different combinations of type definition for incoming and outgoing flows of processes of two-hemisphere model. At the moment authors try to investigate the possibility of exact definitions of method's arguments based on information in two-hemisphere model. A deeper analysis of notational conventions of nowadays available notations for business process modeling is required and could be stated as one of further researches directions.

## 6. Acknowledgements

The research reflected in the paper is supported by the research grant No. R7389 of Latvian Ministry of Education and Science in cooperation with Riga Technical University "Development of tool prototype for generation of software system class structure based on two-hemisphere model".

## 7. References

[1] Balzer R. A 15 year perspective on automatic programming. IEEE Transactions on Software Engineering. Vol 11, No 11, November, 1985.

[2] Krogstie J. Integrating enterprise and IS development using a model driven approach. Proceedings of 13th International Conference on Information Systems Development – Advances in Theory, Practice and Education. Vasilecas O., Caplinskas A., Wojtkowski W., Wojtkowski W., G., Zupancic J., Wrycza S. (Eds). Springer Science+Business media, Inc. pp.43-53, 2005.

[3] Siegel J. Developing in OMG's Model-Driven Architecture. OMG document omg/01-12-01. http://www.omg.org/mda/papers.htm, 2001.

[4] UML Unified Modeling Language Specification, OMG document / Retrieved from: http://www.omg.org, 2004

[5] Frankel D. Model Driven Architecture: Applying MDA to Enterprise Computing, Woley Publishing, Inc., Indianapolis, Indiana, pp.352, 2003.

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

[7] Mellor S.J., Balcer M.J. Executable UML. A Foundation for Model-Driven Architecture, Addison-Wesley, Boston, pp.368, 2002.

[8] MDA Guide Version 1.0.1, http://www.omg.org/docs/omg/03-05-01.pdf

[9] Pokorny T., The Model Driven Architecture: No Easy Answers. Proceedings of Simulation Conference, http://www.consec.com.au/simtect2005/abstracts.html?confid=STT2005, 2005 .

[10] Nikiforova O., Kirikova M., Pavlova N. 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", O. Vasilecas, J. Eder, A. Caplinskas (Eds.), pp.291-304, 2007.

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

[12] Gardner T., Griffin C. A Review of OMG MOF 2.0 Query/Views/Transformations Submissions and Recommendations Towards the Final Standards, OMG documents ad/03-08-02, http://www.omg.org, 2003.

[13] MOF, Quert/View/Transformations Submissions. Second Revised submission (2004, submitted by DSTC, IBM, CBOP, OMG document, http://www.omg.org, 2004

[14] MOF 2.0 Query / Views / Transformations RFP, OMG Document: ad/2002-04-10,http://www.omg.org, 2002.

[15] Skersys T., Gudas S. Class Model Development Using Business Rules, Advances in Information Systems Development. Nilsson A.G., Gustas R., Wojtkowski W., Wojtkowski W., G., Zupancic J., Wrycza S. (eds), Volume 1, Springer Science+Business media, pp.203-216, 2006.

[16] Nikiforova O. General Framework For Object-Oriented Software Development Process, Scientific Proceedings Of Riga Technical University, Computer Science, Applied Computer Systems, 3rd thematic Issue, Riga, Latvia, pp.132-144, 2002.

[17] Nikiforova O., Kirikova M. 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, 2004.

[18] Nikiforova O., Kirikova M., Wojtkowski W. Role of models in knowledge transfer during OO software development, Proceedings of the 15th European – Japanese Conference on Information Modeling and Knowledge Bases, Tallinn, pp. 305-320, May 16-19, 2005.

[19] Pavlova N., Ņikiforova O. Formalization of Two-Hemisphere Model Driven Approach in the Framework of MDA, Proceedings of the 9th Conference on Information Systems Implementation and Modelling, Czech Republic, Prerov, pp. 105-112, 2006.

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

[21] Pavlova N. Several Outlines of Graph Theory in Framework of MDA, Advances in Information Systems Development, New Methods and Practice for the Networked Society, Vol. 2, Maguar G., Knapp G., Wojtkowski W., Wojtkowski W.G., Zupancic J. (eds.), Springer Science+Business Media, LLC, pp. 25-36, 2007.

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

[23] Pavlova N., Approach for Development of Platform Independent Model in the Framework of Model Driven Architecture, Ph.D. Thesis, Riga Technical University Press, p.153, 2008.

[24] Cameron, P. Combinatorics: Topics, Techniques, Algorithms, Cambridge University Press, 1994.

[25] Guttman M., Parodi J. Real-Life MDA: Solving Business Problems with Model Driven Architecture. San Francisco, CA: Morgan Kaufmann Publishers, 223 p, 2007.

[26] GRADE tools, GRADE Development Group, http://www.gradetools.com/

[27] GRADE, Business Modeling, Language Guide, INFOLOGISTIK GmbH, 1998.

[28] Kalnins A., Barzdins J., Podnieks K. Modeling Languages and Tools: State of the Art, Proceedings of 2nd International Conference on Simulation, Gaming, Training and Business Process Reingineering, pp.211-214, 2000.

[29] Repa, V. Modelling Business processes in Public Administration,s Advances in Information Systems Development. Nilsson A.G., Gustas R., Wojtkowski W., Wojtkowski W., G., Zupancic J., Wrycza S. (eds), Volume 1, Springer Science+Business media, pp.107-118, 2006.