Modeling of business processes. 1. Overview of models and metamodels

Sigitas Vilkelis [a]
Department of Informatics, Faculty of Mathematics and Informatics
Vilnius University, Naugarduko 9, Vilnius, Lithuania

Received 28 June 2011, accepted 10 September 2011

Abstract. Different aspects of business process modeling are observed and table structure describing business processes was formalized. Methods and categories of modeling, as well as terminology, existing techniques and tools for *Model Driven Architecture* (MDA) are revised and possibilities of their application in transformation of business process models are evaluated.

Citations: Sigitas Vilkelis. Modeling of Business Processes. 1. Overview of Models and Metamodels – *Innovative Infotechnologies for Science, Business and Education,* ISSN 2029-1035 – **2(11)** 2011 – Pp. 9-15.

Keywords: Model Driven Architecture; MDA.
JEL: C88; M15.
Short title: Overview of models - 1.

Introduction

Nowadays, under the influence of IT on business, it is difficult to imagine a successful but completely not computerized business. The influence of IT on computerization of specific business processes is essential and has many forms - from digital data storage to automatization of difficult multi-user business processes. Creation of business models is very important in keeping up with the advancing technologies. They are necessary for a change of system platform. In order to create a successful model, specific business terminology has to be known along with the software terminology. The resulting model is a combination of two different subjects: the business terminology and the abilities of business process modeling tools.

This is the reason why business models are usually created by two groups of people: the ones employed in a specific area who know the details well and programmers creating the software system to implementing the business model. Business analysts receive the requirements of initial business processes from customers and pass them to programmers.

Althrough the phase of requirement clarification employs several people, a fixed business process description structure is required. It is needed to be able to look for similar processes in the same structure when creating models for other customers as well as enabling formal documentation and creation of requirements that should be met.

The technologies have been advancing rapidly, but customers still use simple tables, filling the forms by type or even by hand. Models created this way have much higher error probability and it takes additional time converting them to modern standards processed by a computer. Switching to structures defined by the modern standards would reduce the workload for everyone - from analysts, communicating with customers, to programmers, doing all the software package creation for the same customer. Due to the customer's habit of using tables, there is a need for special instruments designed to convert these tables to processes understandable by computers.

This work is devoted to the literature review containing business process modeling and its application of architecture related to these models. Methods and categories of modeling as well as terminology, existing techniques and tools for *Model Driven Architecture* (MDA) are revised and possibilities of their application in transformation of business process models are evaluated.

1. General formulation of a task

According to the experience of the author, the following problems occur when using table modeling.

1. A process modeled using tables will not avoid making a lot of errors, until actual programming takes place.
2. Such process may not be added to running business management systems.
3. Logical business process errors are difficult to notice in a spreadsheet description.

The first problem may be solved in three steps. First of all,

[a]Corresponding author, email: *sigitasv@gmail.com*

the structure of a table should be defined. Secondly, rules should be set to define the formal table structure and finally, these rules should be written in formal form.

The solution to second problem may be expressed as a creation of a process to transform table-based business processes to ones using standard notation. This enables easier manipulation with these processes in management of software systems. The third problem may be solved by applying the same method as in the second problem. The transformed process may be represented graphically (as two-dimensional or three-dimensional distribution), which would help to avoid logical errors.

These are the general problems occurring in process modeling. By clarifying these problems, the following steps are pointed out for this work.

A. Define the business processes described in the tables.
B. Create a process of transformation of table-based business process to formal business process notation.
C. Create a structure of rules and define the rules that processes described in tables should follow.
D. Demonstrate the process of rules and transformation inspection by releasing a prototype of business process transformation and verification.

In order to achieve the A-task, partial sub-tasks were done.

A1. Analysis of existing business process notations was done and a common set of elements was taken from them.
A2. According to the resulting list, formal structure was made and fields of data as well as data types within fields were described.

The steps to complete the B-task are presented below.

B1. Architectural ideas based on models for transformation of table-based business processes to formal business process notation were revised and applied.
B2. Rules of transformation were defined.

C-task was completed by performing the following steps.

C1. A set of rules to be used by table-based business process was defined.
C2. This set of rules was applied to formal table-based business processes.

Actions taken to accomplish the D-task were described in three positions.

D1. According to the business modeling at a workplace of the author and analysis of References, entirety of technological tools for practical solution was made.
D2. A tool from a selected technological environment was chosen to ensure the transformation of table-based business process to formal business process notation.
D3. The selected tool was extended with a rule inspection.

The specific steps and their execution are described in following chapters.

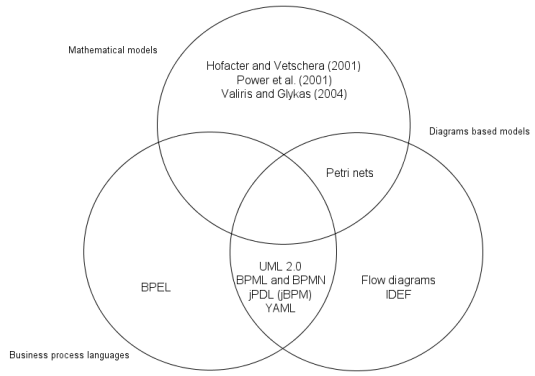

Fig. 1. Classification scheme of the business process modeling. Adapted according to Ref. [1].

2. Methods and notations. Review

Business processes may be modeled using various methods, such as mathematical, chart plotting tools or languages defining business processes. Fig. 1 represents the classification scheme of a business process modeling.

As we can see in the Fig. 1, strict technique separation into three categories is absent, because, some of them have common features. All categories are reviewed below.

2.1. Categories of business process modeling

Mathematical Methods. Mathematical methods are based on strict formalism which results in models which are completely accurate and satisfying all requirements. These models have fewer errors and may be verified using formal methods. Nevertheless, it is inconvenient to create business models using mathematical methods, because advanced mathematical, logical and field-specific knowledge is required.

Chart-based models. The biggest advantage over mathematical methods to chart methods is that they are defined not in mathematical formula, but in graphical charts. This is very important feature, when there is a need to present it to a customer. The business processes are less formal this way, but much more popular.

Languages of business processes. The most popular models are based on languages of business processes. The popularity is based on the applicability of it - all languages can be represented graphically and have their unique document saving format. . Graphical representation is important for understanding the process and the possibility to save them provides a possibility to exchange the documents between different tools.

2.2. Description of business process modeling

Modeling of business processes could be performed using specific tools (created for business process modeling only),

although there are some tools that are general modeling tools with extensions enabling business process modeling. Most of the methods exist as standard tools, such as *Object Management Group* (OMG), *Unified Modeling Language* (UML), *Business Process Modeling Notation* (BPMN), *XML Process Definition Language* (XPDL) [2].

BPEL4WS. Modeling standard BPEL4WS was created by unified forces of BEA, IBM, Microsoft and other companies. The purpose of it is describing of processes provided on Internet and enables creation of difficult business processes, by combining several independent actions to one set of jobs. Such combination is suitable for business process modeling and it is based on *Web Service Definition Language* (WSDL), so different web services exchange XML documents. Service oriented (SO) business process modeling has its own advantages over methods using strict process arrangement. They are: flexibility and easiness of changes [3].

BPMN. BPMN is a standard created by *Business Process Management Initiative* (BPMI), describing Business Process Diagram (BPD) based on diagram creation technique used in graphical representation of business process models. This standard is very popular amongst analysts as well as programmers, because it puts the client needs in a simple way, but it has only the graphical notation, lacking formal description of business process. BPMN documentation [4] also lacks definition of how should the graphical data be stored in a way that would be understandable by computer. This creates an uncertainty of a format used, because every tool uses its own data format (incompatibility could occur). Although, there are some tools that use XPDL's XML documents [5].

There are several important notation problems of BPMN explored in Ref. [6]. This notation has nothing in common with representation of user interface of business process in program system and there is no connection between the business process and the modeled field. The authors of Ref. [6] found a solution to both of these problems. Using rules of transformation, a problem of transforming BPMN diagrams into YAWL diagrams is analyzed in Refs. [7-8]. Working tool is able to perform such transformation, which also shows, that BPMN notation described formally may be transformed into any other formal structure.

UML. UML is one of the most popular program systems and business process modeling languages. It defines many different diagrams made for objective system modeling, and diagrams such as activity enable modeling of fully-fledged business process. Business processes may be defined as an oriented graph, made of peaks and bows using UML. Peaks represent the performed single or combined activity. Peaks may be used for execution control as well. Start and end peaks are assigned to execution control peaks, marking the beginning and the end of a process. Connection and branching in UML may be used to model separation or addition of several parallel processes [2].

UML sequence diagram can be used to show exchanging the information between people participating in business process. Extended activity diagram is suitable for creation of a model describing communication between people [9]. Modeling of variability using UML is presented in Ref. [10]. Common fields where changes are usually made are presented in this paper, these are the changes during the business process, data transmission sequence or activities of business process. It is said, that the description of business process should not go into details, but instead represents only the general sequence of actions making space for actions that could possibly change over time. Nevertheless, this method can only be theoretical, because it is essential to specify common input and output points, sequence of activities, even the exact processed data in practical business process.

The information provided by Ref. [11] describing the transformation of UML business process to XPDL business processes is very beneficial. The general idea is to fill in the missing data in UML diagrams and transform them by using XSLT to XPDL. In order to have XPDL documents fully structured, UML diagrams are filled with extra stereotypes.

XPDL. XPDL is a language of business process description introduced by *Workflow Management Coalition* (WfMC) to define a general data exchange form and supporting moving of different process description between different tools. The purpose of XPDL 2.0 is description of business processes presented in BPMN graphical notation.

Petri net. *Petri net* is a way of business process modeling using mathematical methods and graphical imaging. It consists of places, transitions and arcs. Places may be marked and moved to other places by following the rules. It is very convenient to describe and analyze parallel, asynchronous or distributed systems. As a graphical tool, Petri net can be used to represent graphical connections similar to sequence diagrams. Simulation of dynamic and parallel systems is also performed by using bookmarks in these Petri nets.

The example of business process modeling is presented in Ref. [12]: the basic of modeling is based on two components - activities and resources. Specific resources, such as human labor, specific data or even Internet services, are required for these activities. The biggest benefit of using Petri net is the scalability of models: using several layer modeling, even the smallest processes may be modeled and combined to larger ones, thus creating a clear and detailed business model.

Another similar sample of modeling is described in Ref. [13]. The main difference between the previous one is the usage of several Petri nets: ontological, based on abstract understanding of business process, concept, introducing the

business transaction term and system interfaces and functional net describing interfaces, services and data streams. This type of model enables the evaluation of a business process from three different angles, which proves useful when there is a need to confirm the logic of a process.

IDEF0 and IDEF3. The purpose of *Integration Definition for Functional Modeling* (IDEF0) is functional modeling based on usage of text and graphical markings on organized and systematic models. This increases the understanding of project and integration activities as well as defines the requirements.

Method called *Integrated DEFinition for Process Description Capture* (IDEF3) defines the collection and documentation of processes. It incorporates easy-to-understand priorities and connections between actions.

VPML-S. A new graphical business process modeling language *Service-Oriented Visual Business Process Modeling Language* (VPML-S) was created as a language based on UML, extended with stereotypes [14]. Its purpose is modeling of service-based business processes. The main goal was to create a language that would have a decent graphical notation and would not require specific IT knowledge in order to use it. A business process written in this language is fully compatible with BPEL language, which defines business process as a septum: activities, products, resources, connections, events, attributes, partners. Every part of the septum is strictly defined using mathematical methods. It may be stated that it can be used to model business processes because it supports Internet services, although there are a couple of problems - there may be a lack of support for the language and poor availability of tools enabling modeling in this language, because it is fairly new, written in 2008 and designed for academic purposes.

Existing Activity Diagram. Modeling of a Business Process could be based on *Existing Activity Diagram*. If a business has documentation of its activities as activity diagrams, they may be transformed to business processes as if reusing them [15]. Such modeling of business processes saves time on analysis and documentation of existing business processes. It may also increase the quality of business processes and reduce the error probability. The information on usage of such diagrams should be retrieved at first, in order to know whether they are still valid: for example, if a company kept records of such diagrams for their first year, but discontinued afterwards, there is no use for these diagrams.

JBPM and JPDL. *JBoss Business Process Management* (JBPM) is a management system, filling the gap between analysts and programmers. It is flexible and provides a way of process modeling, suitable for both of these groups.

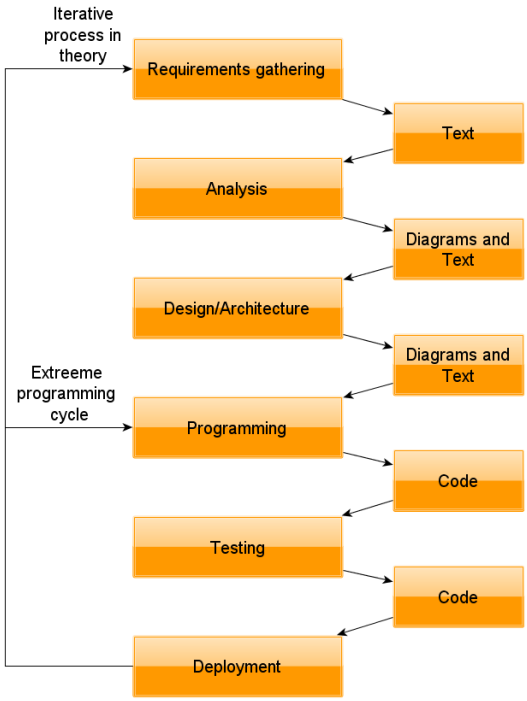

Fig. 2. Traditional cycle of software creation. Adapted according Ref. [16].

Input data for JBPM is presented as descriptions of graphical business processes. The process represents a sequence of actions that are defined as transitions from one activity to another. These graphical diagrams of business processes are the basic way of communication between analysts and programmers.

JBPM as JBPM Process Definition Language is based on *Process Virtual Machine*, which is able to support several languages devoted to the business process definition. JPDL is currently the basic language, created by a business itself. JPDL is a flexible language with extension possibilities, which, according to experience of author, enables easy implementation of JBPM JPDL processes to active systems [16].

3. Transformation technique

3.1. Models and meta-models

The amount of tools and techniques for digitization of businesses has been growing constantly as well as the amount of digitized businesses themselves. One of the systems was based on creation of a model for every step describing it with a required level of detail. This method prevents creation of many documents and allows transformation of the model to a certain software. Such technique was named *Model Driven Software Development* (MDSD). Traditional cycle of software creation is shown in Fig. 2, and extended cycle of *Model Driven Architecture*(MDA) based software creation is shown in Fig. 3. Several software development areas are based on MDA. One of them is called *Model Driven Engineering* - MDE.

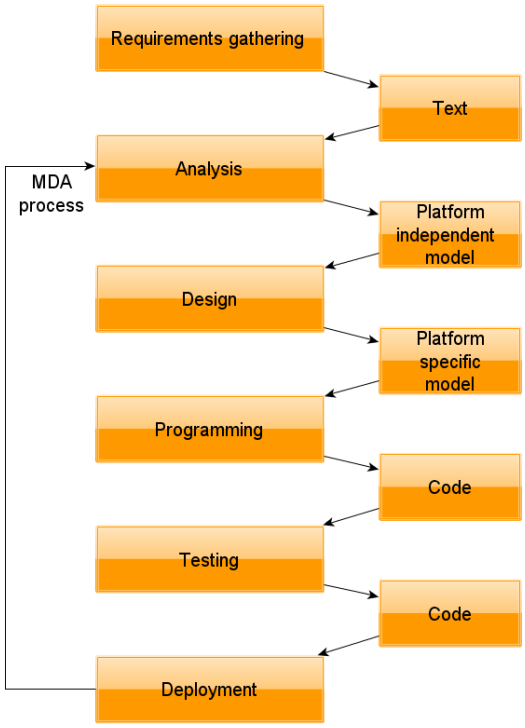

Fig. 3. MDA based software cycle creation. Adapted according Ref. [18].

MDA is developed by OMG group and it specializes not only in software development but in separating logic of business and software from a specific technology in software. A digitized business solution defined by *Platform Independent Model* (PIM) based on UML and other OMG standards with MDA ideas may be implemented in any specific platform using web services, such as .NET, J2EE and others. PIM models define a specific software functionality required by business separately from a software based on specific technology.

That way technological restrictions are avoided and moving to another technological environment freely is encouraged. *Platform Specific Models* (PSM) are derived from PIM models and then transformed to a software supported by specific PSM environment [18-21]. Relation between PIM and PSM is shown in Fig. 4.

One of the general properties of MDA is transformation of models. During one transformation, PIM is joined with additional information and PSM is generated. During another transformation, a realization of software is generated using mapping method. A realization of specific transformation depends on the software system. A kind of transformations exists when models written in PIM language are transformed to models written in PSM language. PIM and PSM meta-models and rules of transformation are defined to enable such process. This transformation is performed between two PIM and PSM models with specific values. Graphical representation of this process is shown in Fig. 5.

Fig. 4. Relation between PIM and PSM.

3.2. Types of model transformations

Model-to-Model (M2M) transformations have been presented previously [17, 19]. There is another way of transformations - the so-called *Model-to-Text* (M2T) transformation. It converts a model to any text: from software code to documents of any format. A lot of various tools can be either commercial or open-source. Both types will be reviewed.

Open-Source Model Transformation Tools. *Kermeta* package was created by INRIA Triskell [20]. It is based on Eclipse platform and the environment is of object-oriented type. The purpose of it is describing and transforming of models and meta-models, as well as their simulations. Kermeta is created as an extension to *Eclipse Modeling Framework* (EMF).

MOFScript is a tool for M2T transformations based on EMF as well. Its purpose is transformation of models and metamodels based on Meta-Object Facility (MOF).

The IBM *Model Transformation Framework* (MTF) - is a tool for describing relations between meta-models in QVT and it is based on EMF as well.

The *ATL Engine* - a language similar to QVT written by INRIA Atlas. It is one of the most important technologies in Eclipse M2M project, created as a bunch of add-ons and it works as a built-in programming language to perform, describe and trace transformations between models [22].

OpenArchitectureWare (oAW) - a flexible framework working along XMI and based on templates.

Generative Model Transformer (GMT) - an Eclipse project for a model transformation technology for Eclipse. Several current tools are a part of GMT: AMW (Model Weaving), Epsilon (Model Merging), MoDisco (Model Discovery), MOFScript (M2T), openArchitectureWare, UMLX (Graphical Transformation), VIATRA2 (Visual Automated Transformations).

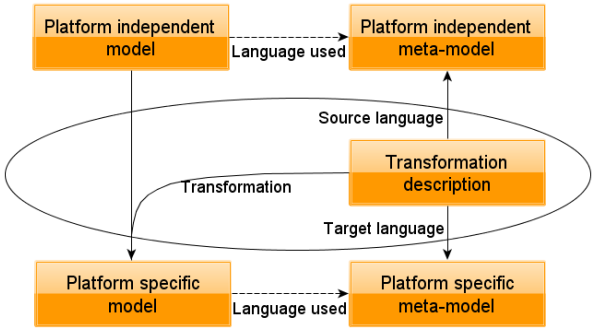

Fig. 5. Process of transformation when using meta-models. Adapted according Ref. [21].

OpenMDX - open-source MDA environment consisting of several tools based on XMI and supporting multi-platform (J2EE, .NET) code generation [21].

Commercial Model Transformation Tools. *ArcStyler* - a commercial MDA tool created by Interactive Objects. It is sold along with MagicDraw UML tool, but it supports other UML tools as well.

Model Component Compiler (MCC) - a commercial product of InferData supporting transformations of M2T to J2EE.

Xactium XMF Mosaic - a tool supporting M2M transformations.

Model-in-Action and MDA - a tool created by Mia software, that supports generation of software and M2M transformations on flexible framework.

MetaEdit+ - built-in environment of modeling and meta-modeling for creation of languages and source generation. It supports XML and SOAP/Webservice transformations for models and meta-models.

MDWorkbench - a tool supporting M2T and M2M transformations accepting any meta-model format as an input. It is based on Eclipse and EMF [22].

Conclusions

Business process modeling of author's workplace was analyzed and table structure describing business processes was formalized. A process of transformation of a table-based business process to a standard description business process notation, based on MDA, was defined. Rules that have to be met by table-based business process were structured and described. Transformation and rule checking were implemented using Eclipse with standard and oAW plugins.

A successful prototype and process definition prove that objectives were completed successfully. The achieved results will improve and speed up work of several people: analysts will be able to check and see graphical business process while filling the business process description table. This will prevent logical errors that occur while creating a business process from separate tasks. Programmers will be able to retrieve the notation which are structured and depicted in specific form of business process. Transformations will be done directly from table-based business process. They will only need to fill some additional information and implement it to a running system. The transformation process using BPMN notation has obvious advantages - business processes described by this notation can be transformed to almost any other business process notation providing almost unlimited expandability for such process. This possibility enables company modeling their processes by tables to adjust their final result to used technologies or specific client requests without changing their initial table-based business process transformation to standard description business process.

Acknowledgments

This work was prepared as a part of the Master Thesis. Author would like to thank Prof. Rimantas Vaicekauskas (Vilnius university) for valuable comments and suggestions.

References

1. Kostas Vergidis, Ashutosh Tiwari, Basim Majeed. Business Process Analysis and Optimization: Beyond Reengineering. – *IEEE* 2007, 14p.
2. Wei Wang, Hongwei Ding, Jin Dong, and Changrui Ren. A Comparison of Business Process Modeling Methods. – *IEEE* 2006, 6p.
3. Corine Cauvet, Gwladys Guzelian. Business Process Modeling: a Service-Oriented Approach. – *IEEE* 2008, 8p.
4. Object Management Group. Business Process Model and Notation (BPMN). Version 1.2, 2009-01-03. – < http://www.omg.org /spec/BPMN/1.2/PDF/ >, accessed 2010-05-03.
5. Michael zur Muehlen, Danny T. Ho. Service Process Innovation: A Case Study of BPMN in Practice. – Proceedings of the 41st Hawaii International Conference on System Sciences, 2008, 10 p.
6. Dagmar Auer, Dirk Draheim, Verena Geist. Extending BPMN with Submit/Responsestyle User Interaction Modeling. – *IEEE* 2009. Conference on Commerce and Enterprise Computing, 2009, 7p.
7. JianHongYE, ShiXin SUN, Wen SONG, LiJie WEN. Formal Semantics of BPMN Process Models using YAWL. – Second International Symposium on Intelligent Information Technology Application, 2008, 8p.
8. JianHong YE, ShiXin SUN, Wen SONG, Lijie Wen. Transformation of BPMN to YAWL. – International Conference on Computer Science and Software Engineering, 2008, 6p.
9. Wang Hongxiu, Wang Gang, Wen Xiaoxian, Gao Guoan. Business Process Modeling for Multi-enterprise Cooperation. – *IEEE* 2007, 4p.
10. Maryam Razavian, Ramtin Khosravi. Modeling Variability in Business Process Models Using UML. – *IEEE* 2008, 6p.
11. Ping Jiang, Quentin Mair, Julian Newman. Using UML to Design Distributed Collaborative Workflows: from UML to XPDL. – *IEEE* 2003, 6p.
12. Hongmei Gou, Biqing Huang, Wenhuang Liu, Shouju Ren, Yu Li. Petri-Net-Based Business Process Modeling for Virtual Enterprises. – *IEEE* 2000, 6p.
13. Xin Wang, Yanchun Zhang, Hao Shi. Scenario-based Petri Net Approach for Collaborative Business Process Modelling. – *IEEE* 2007, 8p.

14. Shaomin Xing, Bosheng Zhou, Tianying Chen. A Service-Oriented Visual Business Process Modeling Language. – *IEEE* 2008, 5p.
15. Chao Yu, Guoqing Wu, Mengting Yuan. Business Process Modeling Based on Workflow Model Reuse. – *IEEE* 2005, 4p.
16. JBoss. JBoss jPDL. – <http://www.jboss.org/jbpm >, accessed 2010-05-01.
17. A. Kleppe, J. Warmer, W. Bast. MIDA Explained. The Model Driven Architecture: Practice and Promise. – Addison-Wesley, 2003, 170 p.
18. OMG. OMG Model Driven Architecture. – <http://www.omg.org/mda/>, accessed 2010-05-01.
19. Eclipse Foundation. Model To Model. – <http://www.eclipse.org/m2m/>, accessed 2010-05-01.
20. Eclipse Foundation. Eclipse Modeling Framework Project. – <http://www.eclipse.org/modeling/emf/>, accessed 2010-05-01.
21. Igor Sacevski, Jadranka Veseli. Introduction to Model Driven Architecture (MDA), Seminar Paper. – 2007-06, 15p.
22. Pau Giner, Victoria Torres, Vicente Pelechano. Bridging the Gap between BPMN and WS-BPEL. M2M Transformations in Practice. – 2006, 14p.