

Modeling of business processes.

2. Prototypes of transformation

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. Working method devoted to formal table-based business process transformation using Eclipse to formal business process notation is presented here. The resulting formal business process notation document may be used as a template in further programming jobs by implementing it to a working system.

Citations: Sigitas Vilkelis. Modeling of Business Processes. 2. Prototypes of Transformation – *Innovative Infotechnologies for Science, Business and Education*, ISSN 2029-1035 – **2(11)** 2011 – Pp. 16-21.

Keywords: Model Driven Architecture; MDA; Eclipse.

JEL: C88; M15.

Short title: Prototypes of transformation - 2.

Introduction

Previous publication [1] is aimed to review the different aspects related to transformation between business processes. Methods and categories of modeling, as well as terminology, existing techniques and tools for *Model Driven Architecture* (MDA) were revised [2-4] and possibilities of their application in transformation of business process models are evaluated.

The basics of the business process transformation and requirements of expandability are defined, as well as the input and output data on every step of transformation. A choice of using indirect transformation, by adding a temporary business process notation, instead of direct transformation is explained.

A working method devoted to formal table-based business process transformation using Eclipse to formal business process notation is presented here. The resulting formal business process notation document may be used as a template in further programming jobs by implementing it to a working system.

1. MDA Tools for models

MDA is a new outlook to a software, implementing traditional models by using them as input or output data. The purpose of MDA is replacing traditional diagrams, such as UML [5] or ordinary text by models. This allows access to lower layer models by applying transformations, generating machine code at the lowest layer. All these transformations are defined by specific rules of transformation from one model to another [6].

MDA technique was used in this work to perform a transformation from a business process described in a table to formal business process notation.

According to author's experience, analysts of business systems still use tables instead of standard approved techniques and methods when communicating with their customers to provide easier apprehension throughout computerization of a business. The objective of this work was created to make sure that transformation of structured tables to a digital format understood by computer is possible. This provides several ways to find the solution.

One of them is the transformation of documents described above using XSLT. It is not very convenient, because if the initial table structure changes, which may occur when analyst has to adjust it for different customers, the structure of XSLT transformation has to change too. This may be a problem while working with several customers at once as well, because different XSLT transformations are needed for every separate table.

Another way is to separate variable part of table-based business structure from standard description business process describing the resulting non-variable part. A transformation between table-based business structure and standard description process should be performed by defining meta-models describing it, because standard description process has to always remain unchanged, even if the table-based process changes. The most convenient representation of table-based process structure is a CSV document with a predefined row-column structure.

The standard description business process can be defined using BPMN and a transformation between CSV and BPMN

^aCorresponding author, email: sigitasv@gmail.com

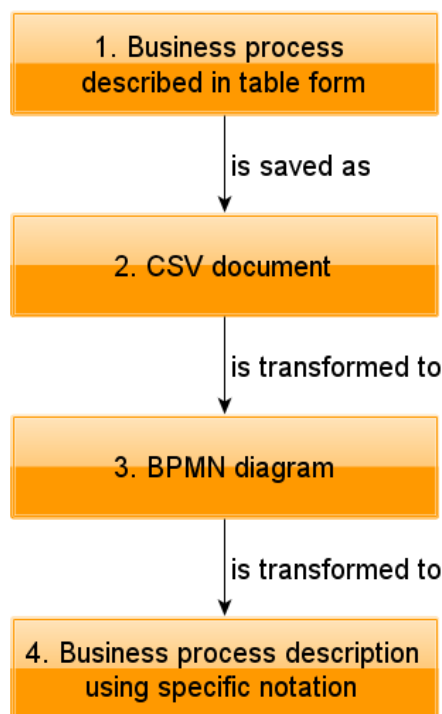


Fig. 1. Transformation of a formal business process table to a specific business process notation.

meta-models may be performed [7].

The BPMN structure may not be applied to a system directly, therefore a transformation to a specific business process notation has to be performed. Fig. 1 represents the transformation process graphically.

A third way to solve the problem of transformations is saving a formal table-based business process as a CSV or any other document format capable of maintaining organized row-column structure. Every section of CSV document is separated by using a comma or semicolon. Several sections constitute a record which takes up one row. A table of business process description in such form may be analyzed and processed by using computer tools.

To avoid scenario presented in first solution, variable and non-variable parts have to be separated. The conversion to specific business process notation is performed after transformation of CSV document to standard business process notation, which is BPMN in this case. As the analysis of literature has shown, business processes in BPMN are much clearer and accessible with many software development tools. The notation itself has a wide choice of available elements ensuring proper conversion to a chosen specific business process notation.

The conversion between CSV and BPMN may be performed by using XSLT transformation as well, although a problem of supporting the platform appears. In order to avoid it, meta-models describing CSV and BPMN business processes should be defined. After that, transformation rules have to be

introduced as well. Another part of the process is transformation of BPMN to a specific business process notation. They should be chosen according to the experience of maintenance personnel and current software running in a business they are going to be used on and to ease the implementation. The meta-model of specific business process notation, as well as the transformation rules have to be defined.

The process described above, enables the following:

- i) the meta-models and transformation rules between BPMN and specific business process notations may remain constant when the business processes defined in CSV change;
- ii) changing the transformation rules independently of CSV, BPMN or specific business process notation meta-models;
- iii) transformation of business process defined in BPMN to any specific business process notation, so changing platforms may be performed keeping old and working business processes by transforming them to the new notation.

As we can see, the resulting specific business process notation is ready to implement along the modeling tools and technologies used in business, even if it is not completely full because of specifics of different notations.

Still, the resulting notation may be a decent fundament for further implementations and it minimizes the amount of human labor for conversion of a table-based business process to a specific business process notation. A prototype resembling this transformation will be presented in next chapter.

2. A Prototype of Business Process Transformation

A prototype of transformation of formal table-based business process to a standard description business process was released by using a plentiful list of literature and the theoretical part revised above.

2.1. General Process of Transformation

A decision to perform an experiment of transformation of CSV document to a specific business process notation using a prototype was made. The specific business process notation was selected to be a notation based on JBPM JPDL standard used in author's workplace. The graphic representation of the bonds between models and data are shown in Fig. 2.

It is based on M2M transformations which are described by standard and the oAW plugins of Eclipse [8-9]. The crucial transformation is between CSV Ecore model and the Ecore model of general BPMN elements used for other standards. When the requirements of the prototype were known, open-source Eclipse environment was chosen, supplemented with standard and oAW plugins:

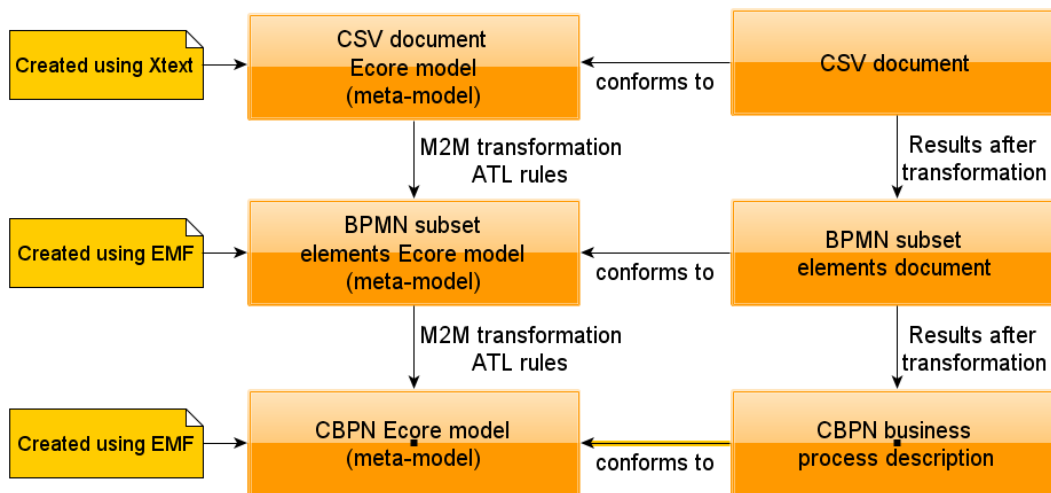


Fig. 2. Bonds between models and data used in transformation.

- i) EMF/Ecore was used for description and securing of meta-models;
- ii) ATL - a plugin supporting transformation language enabling creation and execution of transformation rules [10];
- iii) oAW Check - a tool for description of rules of meta-models defined by Ecore construction, enabling the confirmation whether a specific meta-model follows the rules.

The graphical environment of technology is shown in Fig. 3.

At first, the structure of CSV document was analyzed and *Domain Specific Language* (DSL) was defined using the Xtext plugin for Eclipse. The DSL was used to get an Ecore model, describing the structure of CSV document in terms of Eclipse EMF framework. Ecore is a meta-model of EMF framework, supporting saving of models in XMI format. XMI is a standard for XML Metadata Interchange between different systems created by OMG.

Because ATL supports M2M transformations between Ecore models, a requirement of having both models in Ecore format was made [10].

BPMN Ecore model was defined afterwards. In order to do that, a standard describing BPMN was analyzed and a set of usable elements was chosen. This set is a subset of all possible BPMN notation elements and an Ecore model was created for it.

Once two Ecore models were obtained, rules of transformation between CSV Ecore and BPMN Ecore models were defined using ATL tool. This was an important step of all conversion process confirming that formal table-based business process can be transformed to a BPMN. Afterwards, final step - transformation to a specific JBPM JPDL business process notation took place.

This transformation is required to make sure that the transformations are correct by comparing the result to other business processes used in author's workplace. Such business

process would be used in further programming and implementation work as well. Afterwards, analysis of the resulting process was performed and an Ecore model was created. Transformation rules had to be defined as well. The final result of this process is an XMI document describing the JBPM JPDL business process of author's workplace. Graphical process of transformation is shown in Fig. 4.

A detailed overview on the process of transformation is shown next.

1. The sample business process is formed as a CSV document.
2. An ATL transformation is performed on it to get a document of XMI format corresponding to the CSV Ecore model.
3. Rules of checking CSV XMI document were formed by using oAW Check Constraints tool for Ecore models.
4. Using this tool, CSV XMI document is statically checked if the initial document has any errors.
5. Afterwards, the CSV XMI document is transformed to BPMN XMI document by using ATL transformation rules.

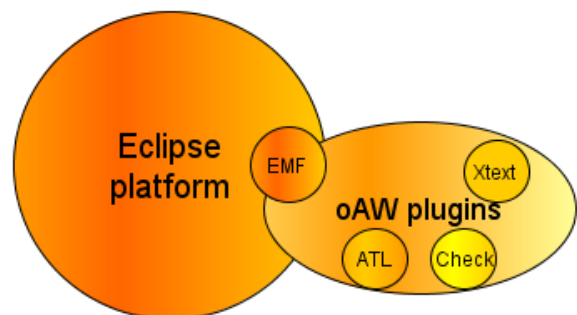


Fig. 3. Technological environment of prototype implementation.

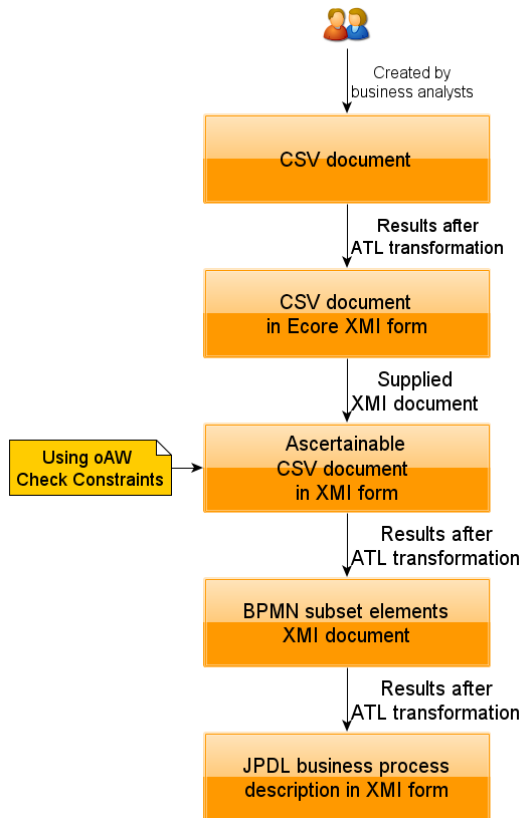


Fig. 4. General transformation process realized in prototype.

The final transformation of BPMN XMI document to JBPM JPDL business process notation is performed, resulting in a XMI document of specific business process notation. A prototype based on Eclipse plugins was realized. The Ecore models and transformation rules used will be presented.

2.2. Ecore Model of CSV Document

A structure of text-based business processes was specified in order to describe a meta-model of CSV document. Formal analyzing of table structure must be done. After analysis of a table structure used for business process notations, a new, more simple and more convenient structure was created, which is shown in Fig. 5.

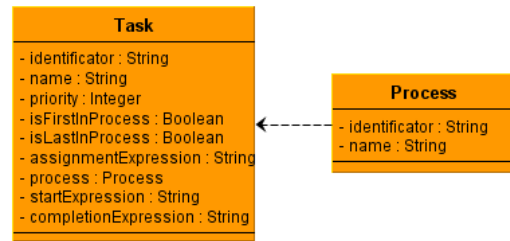


Fig. 5. Metamodel of table-based business process.

The description of fields of formal structure is shown in Table 1.

Rules of business process description table. The data presented in a table of the last chapter have to meet these conditions.

1. Task number field has to be unique for every process.
2. Task name has to be unique in a single process.
3. Task priority has to be between 1 and 5.
4. Every task except last has to have their initial or creation conditions.
5. Task end condition of the last task should not have any other tasks.

These are the basic and initial formal table-based business process rules. The advanced rules that need to be met by business process descriptions are presented below.

1. If a task is not the first one, its initial conditions have to be the same as the end condition of a previous task.
2. If there are several beginning tasks in a process, their initial conditions cannot overlap which means that several initial conditions cannot be met by the same set of data.
3. A process must have at least one first and last task.
4. The same task may not be used in several processes.
5. Initial (beginning) and final conditions may not overlap with one task.

As we can see, every rule is for initial or end conditions because the most of errors are found in these parts. With the help of successful implementation of verification of these parts, the time it takes to model these processes may be reduced significantly.

Table 1. Description of fields of the formal structure.

Table field	Description
Task number	Unique task number for every process.
Name or number of a process	Grouping of tasks to a process.
Task name	Unique task name in a process.
Priority	Task execution priority, a number between 1 and 5.
Initial/task creation conditions	Conditions describing the creation of a task. Comparison of initial attributes with fixed values in a condition.
Task completion conditions	Conditions upon which the current task should be finished and another task created. If there are no more tasks, the process itself is finished.
Is there another task in queue? (Y/N)	Marks whether this is the last task of a process.
Is this task the first one? (Y/N)	Marks the first task of a process.
Expression of task assignment to a user.	An existing username used to login to system is set as an expression value.

```

grammar org.xtext.example.\
CsvDsl with org.eclipse.xtext.common.\
Terminals generate csvDsl "CsvDsl" \
Model : \
(string+=string)*; \
string : \
    number_of_Task = ID'; \
    processName = STRING'; \
    taskName = STRING'; \
    conditionOfCreation = STRING'; \
ExpressionOfDefinitionTo = STRING'; \
    runtime = STRING'; \
    priority = INT'; \
finalysingConditions = STRING'; \
isTheLastTaskInProcess = ('Yes' | 'No')'; \
isTheFirstTaskInProcess = ('Yes' | 'No')'; \

```

Fig. 6. Example of the textual business process DSL.

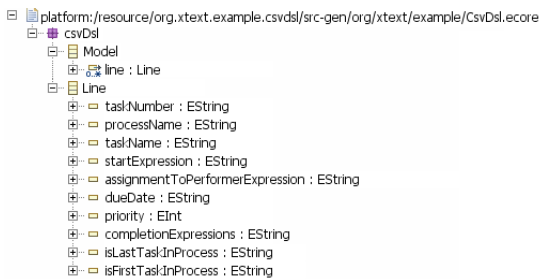


Fig. 7. Ecore model of text-based business process.

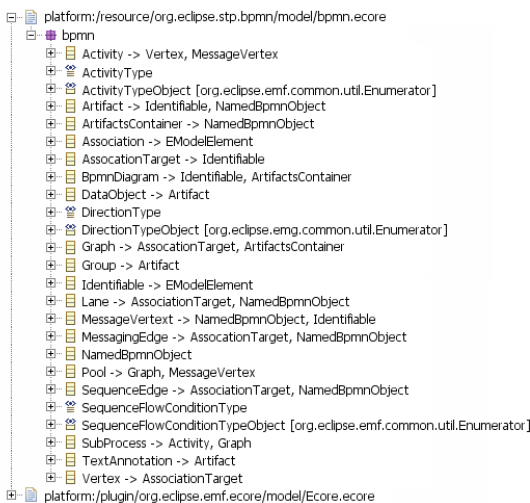


Fig. 8. Ecore model of BPMN elements.

Meta-model of CSV Document. DSL of a business process described using Xtext plugin is shown in Fig. 6. Even complex business processes may be described in such simple structure. Fig. 7 represents an Ecore model created from this DSL using Eclipse plugin. This CSV document model will be used in further M2M transformation process, where this model will be input (source), and standard description business process notation will be transformation output (objective) model.

3. Ecore Model of BPMN Elements Subset

Fig. 8 represents a set of elements and its Ecore model resulted after performing BPMN analysis. BPMN Ecore model was defined using BPMN standard [11]. This BPMN Ecore model is used for semantic BPMN. It is obvious that there should be another model involving depicting of elements (fonts, font sizes, element colours etc.) in business process diagrams, because BPMN is a graphical business process notation. A BPMN plugin using Eclipse platform, creates two documents at once, when creating BPMN business process diagram. One of them is semantic model, while the other is a document containing graphical element parameters. Since graphical positioning of elements is not important, it will be left aside and only semantic document and its structure will be analyzed.

3.1. JBPM JPDL Business Process Notation Ecore Model

The final model transformation result is a JBPM JPDL business process notation document, recognizable by Eclipse plugins. Fig. 9 represents Ecore model describing the document structure for this transformation.

By performing M2M transformation on BPMN Ecore model, a resulting JBPM JPDL Ecore model with XMI document format is received and used in further programming stage for final implementation to current system. Rules of transformation predicting details of the process have to be defined. These details consist of representation of an element from one model to another.



Fig. 9. Ecore model of JBPM JPDL notation elements.

3.2. Transformation Rules

The Ecore models described earlier do not perform any functions themselves, so transformation rules for transformation between models have to be defined. The following M2M transformations were described by using ATL tool and its Eclipse plugin:

- i) from CSV Ecore to BPMN Ecore models;
- ii) from BPMN Ecore to JBPM JPD L notation Ecore models;

ATL tool enables definition of transformation rules between a source Ecore model and a objective Ecore model bonding elements between them. An example transformation rule for converting a business process name from CSV Ecore to BPMN Ecore model is shown in Fig. 10.

```
rule ProcessName {\
  from\
  s : CsvDsl!Line\
  to\
  t : bpmn!BpmnDiagram (\
  title <- s.processName\
}
```

Fig. 10. An example of a transformation rule.

The transformation rules were defined and a JBPM JPD L business process notation XMI document was received. This experiment shows that a business process described by a formal structure table may be converted to a business process notation of standard description type.

References

1. 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.
2. Igor Sacevski, Jadranka Veseli. Introduction to Model Driven Architecture (MDA), Seminar Paper – 2007-06, 15p.
3. Wei Wang, Hongwei Ding, Jin Dong, and Changrui Ren. A Comparison of Business Process Modeling Methods – *IEEE* 2006, 6p.
4. Wang Hongxiu, Wang Gang, Wen Xiaoxian, Gao Guoan. Business Process Modeling for Multi-enterprise Cooperation – *IEEE* 2007, 4p.
5. Maryam Razavian, Ramtin Khosravi. Modeling Variability in Business Process Models Using UML – *IEEE* 2008, 6p.
6. A. Kleppe, J. Warmer, W. Bast. MIDA Explained. The Model Driven Architecture: Practice and Promise – Addison-Wesley, 2003, 170 p.
7. Dagmar Auer, Dirk Draheim, Verena Geist. Extending BPMN with Submit/Responsestyle User Interaction Modeling – *IEEE Conference on Commerce and Enterprise Computing*, 2009, 7p.
8. Eclipse Foundation. Model To Model – <<http://www.eclipse.org/m2m/>>, accessed 2010-05-01.
9. Eclipse Foundation. Eclipse Modeling Framework Project – <<http://www.eclipse.org/modeling/emf/>>, accessed 2010-05-01.
10. OMG. ATL – Atlas Transformation Language – <<http://www.eclipse.org/m2m/atl/>>, accessed 2010-05-01.
11. 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.

Results and Conclusions

Successful achievement of goals and tests of prototype leads to following statements:

- i) formal structure tables may be used instead of standard business process notations to describe business processes;
- ii) formal structure tables can be transformed to standard description business process notations;
- iii) created prototype enables automation of conversion between formal table business process and a standard description business process.

The created solutions and achieved results may be applied in any business where business processes are not modeled using standard notations. By applying presented ideas and structuring initial business process tables to a structured row-column structure, transformations to a standard business process notation may be performed.

The described solution may be improved by expanding business process table structure, as well as increasing amount of fields moved from it to a middle BPMN notation, as well as defining transformation rules for several business process notations instead of one. That way, business processes could be transformed to more business process notations, as well as the sketches received after transformations would be fuller.

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.