AUTOMATYKA • 2011 • Tom 15 • Zeszyt 2

Antoni Ligęza*, Tomasz Maślanka*, Krzysztof Kluza*, Grzegorz Jacek Nalepa*

Modeling BPMN Diagrams within XTT2 Framework. A Critical Analysis**

1. Introduction

Design, analysis and management of progressively more complex *Business Processes* require advanced methods and tools. Two recent approaches to modeling such processes have recently gained wide popularity. These are the *Business Process Modeling Notation*, or BPMN for short [8], and *Business Rules* (BR) [7]. Both of the approaches are mutually complementary and apart of similar application area offer in fact distinctive features enabling process modeling [1].

BPMN constitutes a set of graphical symbols, such as links modeling data flow, various splits and joins, events and boxes symbolizing data processing. It constitutes a transparent visual tool for modeling complex processes promoted by OMG. What is worth underlying is the expressive power of current BPMN. The workflow, however, is of purely *operational nature*, which makes attempts at more formal analysis problematic.

Business Rules offer an approach to specification of knowledge in *a declarative* manner. The way the rules are applied is left over until it comes to rule execution. The rules themselves can play different roles in the system, including inference of new facts/knowledge and inference control.

Note that the two approaches are to certain degree complementary: Business Rules provide declarative specification of domain knowledge, which can be encoded into a visual BPMN model. On the other hand, a careful analysis of a BPMN diagram allows for extraction of certain rules covering the business. Unfortunately, there is no standardized methodology for integration of BPMN and BR, and in the existing approaches the integration details are not well specified.

The main common problem of BPMN is lack of a *formal declarative model* defining precisely the logic behind the diagram. Thus, formal analysis of such processes is problematic. A specification of Business Processes faces also several other problems, such as the

* AGH University of Science and Technology, Krakow

** The paper is supported by the BIMLOQ Project funded from 2010–2012 resources for science as a research project.

semantic gap between the design and its implementation, as well as lack of dedicated tools for defining the low level Business Processes (BP) logic.

In this paper, preliminary results of the research concerning Business Processes and Business Rules are presented. The approach consists in modeling BPMN process, transforming it into the XTT2 knowledge representation of Business Rules, and executing in the HeaRT rule engine using particular data from a Database Management System (DBMS).

2. BPMN: a quick overview

BPMN [8] is a visual notation for modeling business processes. The notation has been developed by the Business Process Management Initiative (BPMI) and currently is maintained by the Object Management Group (OMG).

Although the notation is relatively young, it is becoming increasingly popular. According to OMG, there are more than 70 BPMN implementations of various BPMN tools[1]. The current version of the notation, BPMN 2.0, provides several models to cover various aspects and precisely describes the business process. It defines such models as:

1. *Process Model* – describes the ways in which operations are carried out to accomplish the intended objectives of an organization. The process can be modeled on different levels of abstraction: *public* (collaborative Business 2 Business Processes) or *private* (internal Business Processes).
2. *Choreography Model* – defines expected behavior between interacting business participants.
3. *Collaboration Model* – which can include Processes and/or Choreographies, and provides a Conversation view (which specifies the logical relation of message exchanges).

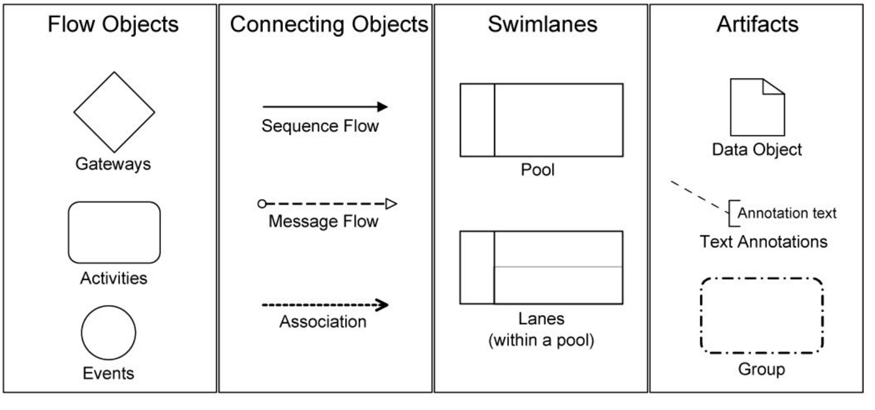


Fig. 1. BPMN core objects

[1] Source: http://www.bpmn.org

In the research described in this paper only the internal Business Process Model with Business Process Diagram is considered. There are four basic categories of elements used in such a diagram (see Fig. 1): Flow Objects (Events, Activities, and Gateways), Connecting Objects (Sequence Flow, Message Flow, Association), Swimlanes, and Artifacts.

3. XTT2: formal modeling of business rules

XTT [2] and XTT2 [6] constitute an efficient tool for design and development of rule-based systems. The main idea consists in putting similar rules operating within the same context into specially designed decision tables. A set of such tables forms a network, and the inference control is passed from a table to another one. Knowledge representation is based on *Attribute Logic*.

The main goal of Attributive Logic (AL) [2] is to describe properties of objects or systems. It is based on symbols denoting given object properties (attributes). To describe an object a set of attributes of that objects must be defined and every attribute must be assigned a value.

Attributive Logic alphabet consists of the following sets of symbols: O – objects names symbols, A – names of attributes, D – attribute domains (attribute values), V – a set of variables, Q_l – logic operators and qualifiers, Q_r – relational end equality operators (such as <, ≤, =).

Any fact formulated in Attributive Logic has the following form:

$$attr\,(obj\,) = val \tag{1}$$

where

$$attr \in A;\ obj \in O;\ val \in D$$

Decision tables, consist of rules, constitute more advanced rule representation. A schematic structure of a table is shown in Table 1.

Table 1

XTT2 table pattern

Rule Id	A_1	A_2	...	A_n	H_1	...	H_k
1	$\propto_{11} t_{11}$	$\propto_{12} t_{12}$	...	$\propto_{1n} t_{1\,n}$	h_{11}	...	h_{1k}
2	$\propto_{21} t_{21}$	$\propto_{22} t_{22}$	...	$\propto_{2n} t_{2\,n}$	h_{21}	...	h_{2k}
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
m	$\propto_{m1} t_{m1}$	$\propto_{m2} t_{m2}$	...	$\propto_{mn} t_{mn}$	h_{m1}	...	$h_{m\mathrm{k}}$

Columns of the table belong to two groups/sets: first, $A = \{A_1, A_2, ..., A_n\}$ defines precondition attributes, second, $H = \{H_1, H_2, ..., H_n\}$ specifies decision (conclusions) attributes. Thus, table schema is $A \rightarrow H$. Each row of the table represents a single production rule of the form shown by (2).

$$(A_1 \propto_1 t_1) \wedge (A_2 \propto_2 t_2) \wedge \ldots \wedge (A_n \propto_n t_n) \rightarrow h_1 \wedge h_2 \wedge \ldots \wedge h_k \tag{2}$$

Symbol α_i denotes a relational operator, and $t_i \in D$ denotes some attribute A_i domain subset.

EXtended Tabular Trees version 2 (XTT2) is a knowledge representation that incorporates the attributive table format [6]. In this approach, similar rules are grouped within separated tables, and the system is split into a network of such tables representing the inference flow. This visual table-based representation can be automatically transformed into HeKatE Meta Representation (HMR), which is suitable for direct execution by the HeKatE RunTime (HeaRT) [4] inference engine. HeaRT also provides a verification module – HeKatE Verification and Analysis (HalVA).The module implements simple debugging mechanism that allows tracking system trajectory, and logical verification of models.

4. Mapping BPMN into XTT2

The idea of a solution of Business Logic implementation and execution is the following. Business Rules are managed by some General Rulebook System (GRBS) application. The rules determine flow of Business Process (actions and theirs order is specified by a BPMN diagram). Next, the BPMN diagram is transformed to Prolog rules, simultaneously Structural Business Rules determine a Database Schema. Prolog rules and Database Schema form a technical description of given Business Logic, which may be next executed within the Platform, which is Prolog Inference Machineand DBMS. The overall idea of the solution (depicted in Fig. 2) may be considered as some general concept of Business Logic implementation using Data Base Management System and Prolog Inference Machine.

This concept includes several areas for further possible research and development, e.g. Business Rules modeling, which is related to General Rulebook System topic that includes managing of natural language expressions. On the other hand, the second area that exists is Business Process modeling, which includes BPMN notation.

The proposed solution bases on using the XTT2 Business Rules that determine flow of Business Processes, specified using the BPMN notation. According to the proposed approach, selected BPMN objects are mapped to rules, which can precisely define the semantics of these elements. The logic of BPMN tasks can be manually defined using rule tables or networks.

The approach uses the BPMN-to-XTT2 translator. The tool has been developed to provide a translation from the XML format of the BPMN model to the XTT2-based Business Rule representation. Moreover, the approach uses several existing BPMN and RBS tools, such as: BPMN modeling tool, HeaRT and HalVA, DBMS. This provides the possibility of executing selected BPMN models in the rule engine. Currently, the translation is provided only for a small subset of the BPMN elements. The diagram sequence flow conditions have to be specified according to the ALSV(FD) logic [6].

In the proposed approach, BPMN elements are translated to the XTT2 rule representation. Thus, each BPD element and its sequence flows are described by the proper XTT2 table filled with rules, which are defined by the logic function appropriate for the particular BPMN element. The detailed specification of the translation and the logic functions for BPD elements can be found in technical report [3]. Below, two examples of such logic functions are presented.

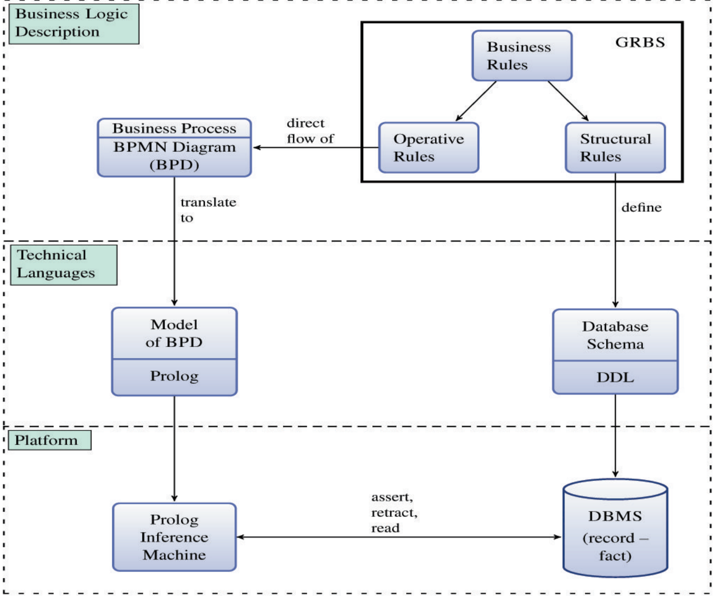


Fig. 2. Business Logic execution concept

Exclusive gateway split $g \in \mathcal{G}_s^{ex}$

$$in(g) = \{f_0\},\ out(g) = \{f_1, f_2, ..., f_n\}$$
$$Cond(f_i) = c_i \quad i \in [1, n-1]$$

`precondition:` f_0

`postcondition:` $f_1 \veebar f_2 \veebar ... \veebar f_n$

`logic function:`

$$f_1 \leftarrow f_0 \wedge c_1$$
$$f_2 \leftarrow f_0 \wedge \neg f_1 \wedge c_2$$
$$\vdots$$
$$f_n \leftarrow f_0 \wedge \neg f_1 \wedge ... \wedge \neg f_{n-1}$$

`notes:` f_n is default path;
$\veebar$ symbol is used to denote xor logic operation.

Throw event $e \in \mathcal{E}^{th}$

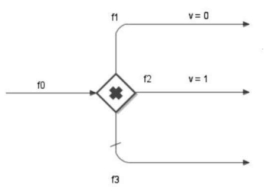
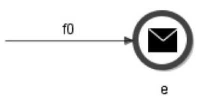
	$in(e) = \{f_0\}$, $out(e) = \{f_1\}$
`precondition:`	f_0
`postcondition:`	$e \wedge f_1$
`logic function:`	$e \leftarrow f_0$
	$f_1 \leftarrow f_0$
`notes:`	throw event is set to true immediately after its input sequence flow is activated (set to true).

In the examples, preconditions denote if the logic of an entity may be executed, while postconditions describe the state of BPD after execution of this entity logic. The logic functions define the entity logic respecting the precondition and postcondition.

Since BPMN does not specify the logic of particular tasks, currently it has to be implemented manually. In the proposed approach, it can be specified either using Business Rules in the form of the XTT2 table or network, or as a HeaRT callback (rule action). Diagram elements are translated to the XTT2 form according to the logical description presented above. Thus, each BPD element and its sequence flows are described by the proper XTT2 table filled with the rules for the particular BPMN element. Examples of XTT2 tables for exclusive split gateway and throw event are presented in Table 2.

Table 2

BPMN to XTT2 transformation examples

<table>
<tr><th>BPMN Element</th><th>XTT2 Table</th></tr>
<tr><td colspan="2">Exclusive split gateway $\mathcal{G}_s^{ex}$</td></tr>
<tr><td>
</td><td><table>
<tr><td>f_0</td><td>v</td><td>f_1</td><td>f_2</td><td>f_3</td></tr>
<tr><td>$= 1$</td><td>$= 0$</td><td>$:= 1$</td><td>$:= 0$</td><td>$:= 0$</td></tr>
<tr><td>$= 1$</td><td>$= 1$</td><td>$:= 0$</td><td>$:= 1$</td><td>$:= 0$</td></tr>
<tr><td>$= 1$</td><td>$\notin [0, 1]$</td><td>$:= 0$</td><td>$:= 0$</td><td>$:= 1$</td></tr>
</table></td></tr>
<tr><td colspan="2">Throw event $e \in \mathcal{E}^{th}$</td></tr>
<tr><td>
</td><td><table>
<tr><td>f_0</td><td>e</td></tr>
<tr><td>$= 1$</td><td>$:= 1$</td></tr>
</table></td></tr>
</table>

The BPMN-to-XTT2 translator is implemented in the Prolog programming language. The acceptable input is an XML-based BPMN diagram serialization exported from the BPMN editor, e.g. Eclipse STP BPMN modeler.

5. Critical analysis

The ongoing research on integration of Business Processes with Business Rules has resulted in several publications [1, 9, 5]. Unfortunately, they do not provide any formalized specification of implementation. Moreover, they do not provide any tool for the solution, but only describe very broadly the topic, and often do not concern the BPMN notation.

There are several tools which provide modeling Business Processes and Business Rules, such as Corel iGrafx Process, IBM WebSphere Business Modeler Advanced, Business Process Visual Architect, or Drools. However, none of them supports executing the Business Processes using a rule engine which supports advanced rule representations like decision tables or trees.

The preliminary results of our research, in contrast to the existing tools, much differ in terms of aims and scope. The presented solution uses rules for specifying both the semantics of BPMN elements and the logic of tasks in the process. Furthermore, these rules are directly used in the execution process. They can also be formally verified.

In order to model Business Processes and Rules in a more comprehensive way, future work includes extending one of the existing BPMN tools by integrating it with the HeKatE Qt Editor (HQEd), a dedicated XTT2 rule editor. Moreover, the unified methodology of BPMN modeling supported by XTT2-based BR will be developed. This will allow for verification and quality assurance of BPs.

6. Concluding remarks

In this paper, some preliminary results concerning an approach to modeling BPMN diagrams with Business Rules using the XTT2 model are presented. The approach consist, in transforming a BPMN process diagram into the XTT2 knowledge representation of Business Rules, and executing in the HeaRT rule engine, using particular data from a Database Management System (DBMS). The approach uses XTT2 Business Rules and the HeKatE rule engine (HeaRT).

The primary goal of the research is to enable automatic translation of BPMN diagrams to declarative, rule-based model. In such a way one can run the HeaRT inference engine for selected models created using Business Process Modeling Notation. Transformation of BPMN diagrams to formalized XTT2 models allows also for formal analysis and verification of its correctness and consistency, as in the case of Rule-Based Systems [2]. Several further dimensions, such as *expressive power* (declarative specification of rules), *modeling parallel/conditional inference* (multithreads), *inference control*, *suitability for formal analysis*, and *robustness* are planned to be investigated.

References

[1] Charfi A., Mezini M., *Hybrid web service composition: Business processes meet business rules*. In Proceedings of the 2nd International Conference on Service-Oriented Computing, ICSOC '04, pages 30–38, New York, NY, USA, 2004. ACM.

[2] Ligęza A., *Logical Foundations for Rule-Based Systems*. Springer-Verlag, Berlin, Heidelberg, 2006.

[3] Maślanka T., Ligęza A., Kluza K., Nalepa G.J., *BPMN to XTT2 translation proposal*. Technical Report CSLTR 2011, AGH University of Science and Technology, 2011, to be published.

[4] Nalepa G.J., *Architecture of the HeaRT hybrid rule engine*. In Leszek Rutkowski and [*et al.*], editors, Artificial Intelligence and Soft Computing : 10th International Conference, ICAISC 2010: Zakopane, Poland, June 13–17, 2010, Pt. II, volume 6114 of LNAI, pages 598–605. Springer, 2010.

[5] Nalepa G.J., Kluza K., Ernst S., *Modeling and analysis of business processes with business rules*. In J.A. Beckmann, editor, Business Process Modeling: Software Engineering, Analysis and Applications, Business Issues, Competition and Entrepreneurship. Nova Science Publishers, 2011, to be published.

[6] Nalepa G.J., Ligęza A., *HeKatE methodology, hybrid engineering of intelligent systems*. International Journal of Applied Mathematics and Computer Science, 20(1):35–53, March 2010.

[7] OMG. *Production Rule Representation RFP*. Technical report, Object Management Group, 2003.

[8] OMG. *Business Process Model and Notation (BPMN): Version 2.0 specification*. Technical Report formal/2011-01-03, Object Management Group, January 2011.

[9] M. zur Muehlen, Indulska M., Kittel. K., *Towards integrated modeling of business processes and business rules*. In 19th Australasian Conference on Information Systems ACIS 2008, Christchurch, New Zealand, December 2008.