“SILVER BULLET” FOR BUSINESS PROCESSES MODELING AND EXECUTION?

Tomislav Rozman, Romana Vajde Horvat, Ivan Rozman
Institute of Informatics
Faculty of electrical engineering and computer science
University of Maribor
Complete address: Smetanova 14, Maribor, SI-2000, Slovenia
Phone: (+386) 2 235 51 07 Fax: (+386) 2 235 51 34

E-mail: {tomi.rozman | romana.vajde | i.rozman} @uni-mb.si

Abstract - The article presents a new set of standards under development for business process modeling, executing, deploying, querying and maintenance. These standards are being developed by many well-known IT organizations gathered within Business Process Management Initiative. The focus of this paper is the description of Business Process Modeling Notation (BPMN) and its transformation to execution model (BPML). Example of a process modeled in this notation is also presented. The possibilities for extension of notation using Resource Description Framework are discussed.

I. INTRODUCTION

Imagine a person, who is responsible for business process modeling in organization. He (or she) is often confronted with a fact, which process modeling technique to use, because there are a great number of techniques and approaches for business process modeling available. Those techniques don’t share common process representation, (which can be graphical or textual), the level of formality or the level of ambiguity. Some techniques are easier to learn than others, which mainly depends of their primary purpose: is the modeling technique intended for improvement of communication between people which are involved in the process or is it intended for automated execution. Modeling techniques for different businesses are also incompatible with each other. One of the main reasons for poor compatibility is, they often offer only a subset of concepts, which are not enough for detailed process description. The experiences from all areas of standardization are showing, it is almost impossible to create a standard that will satisfy all of the different needs. This fact can be applied also to process modeling technique. It is hard to find an optimal modeling technique for people and machines, because requirements for such technique are very comprehensive and they are frequently conflicting. Only techniques with formal rules are appropriate for automated execution and simulation. The following view should be also considered when constructing process-modeling technique: people who are documenting and maintaining business processes often don’t think like programmers (algorithmic way).

One of the latest results to unify different requirements for process modeling technique is held by Business Process Management Initiative (BPMI).

II. BPMI ORGANISATION AND ITS WORK

BPMI is independent, non-profit organization of information technology companies, which was founded in the year 2000. Its goal is to develop a set of XML-based standards for process modeling, execution, maintenance and optimization. Their vision is to develop such standards that would enable vendor independent handling with processes on the same way that RDBMS handles data. The following standards are in the development:

- BPML (Business Process Modeling Language). This is markup language for private processes modeling. Its main purpose is for text based, parse-able and executable description of business processes.
- BPMN (Business Process Modeling Notation). This is graphical notation for process modeling and is based on EPC, Petri nets and Activity diagrams (UML). Its main purpose is for human usage. Its specification also defines mapping rules to BPML and BPEL4WS
- BPQL (Business Process Query Language). This language is used for process state querying, deployment and supervision; likewise SQL is used for relational data querying.

Those standards don't replace other similar new standards in the area of business process management, like ebXML, RosettaNet, Xlang, BPEL4WS, but they are rather complementary.

Regardless of the chosen modeling technique, it must cover all of the important concepts for process modeling.

III. PROCESS MODELING CONCEPTS

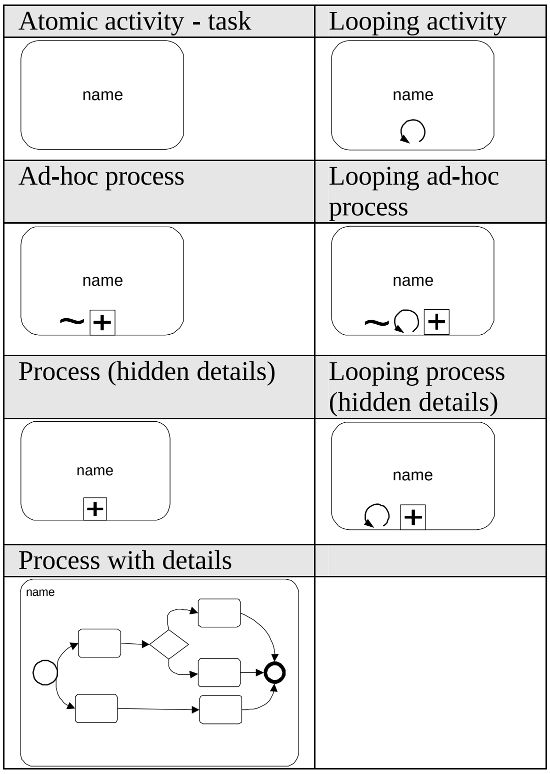
The process modeling technique should support modeling of basic process elements, which include:

- Concepts: activity, event, state, information object
- Patterns: sequence and parallelism, decision, condition, merging, branching, information hiding, process concepts grouping.

Authors of graphical process modeling notation shouldn’t consider only mentioned process elements, but also other aspects, like clarity and readability of the notation. These aspects are not so important in text-based process descriptions, because their primary intention is for automated execution and their primary requirement is unambiguous definition of the process. It can be said that graphical notation BPMN covers mentioned concepts very well.

IV. BUSINESS PROCESS MODELING NOTATION

BPMN can be treated also as a visual representation of Business Process Modeling Language. It is impossible to represent every detail of the process using graphical symbols, therefore every BPMN element can be described using pre-defined attributes [3]. BPMN specification defines graphical symbols for processes, atomic activities, events, grouping of elements using pools and lanes and rules for interconnecting these elements.
At the time of writing this article, the BPMN specification is still in the development and some elements are missing (e.g. synchronization).


Atomic activity - task

Looping activity

Ad-hoc process

Looping ad-hoc process

Process (hidden details)

Looping process (hidden details)

Process with details

Figure 1: Activity types

Sequence of activity execution is represented with full line and message flow with dashed line [Figure 2]

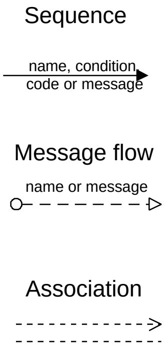
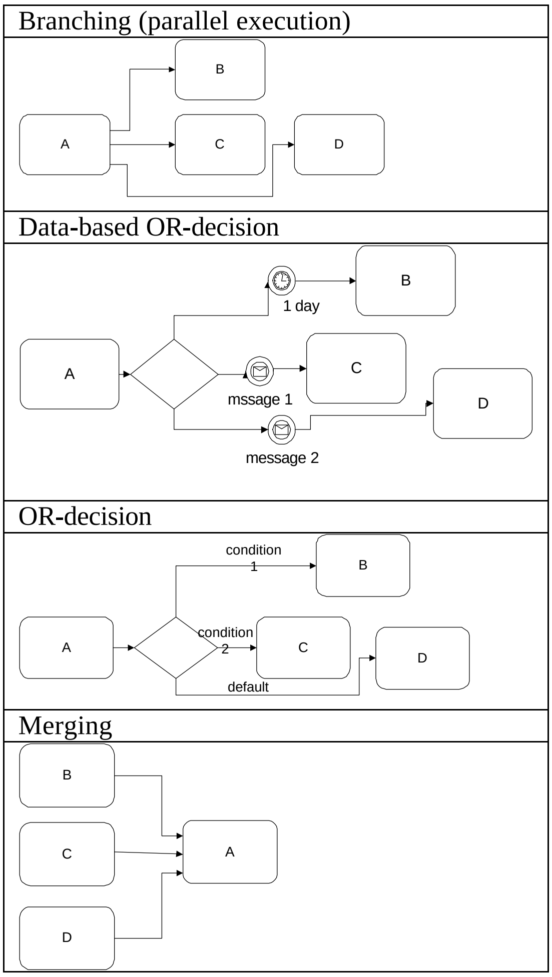


Figure 2 : Association types


Branching (parallel execution)

Data-based OR-decision

OR-decision

Merging

Figure 3 : Branching, decisions and merging

Branching and decisions are modeled as shown in picture [Figure 3]. The difference between mentioned two patterns is, when using branching, all of the successor activities are executed in parallel. Using decision pattern, only one of the successor activities is executed. Decision, which one is executed, can be based on a rule or an event.
A special attention should be put on the combination of usage of multiple branching, conditions and merging patterns. In this case, the part of the process can become ambiguous and therefore inappropriate for mapping to execution language (BPML).
There are tree types of events, which can be modeled using BPMN: start event, stop event and intermediate event. Every type of an event can represent one of the following sub-types of the event: message event, timer event, process error, link event, compensate event, rule-based event or multiple event.

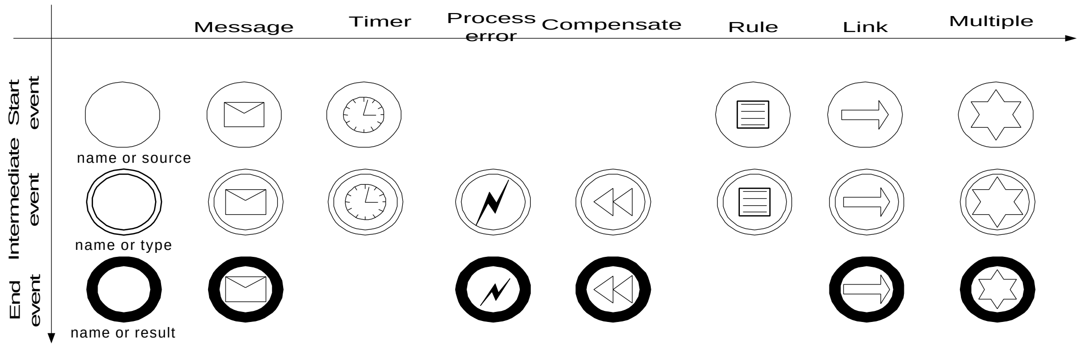


Figure 4 : Event types and sub-types

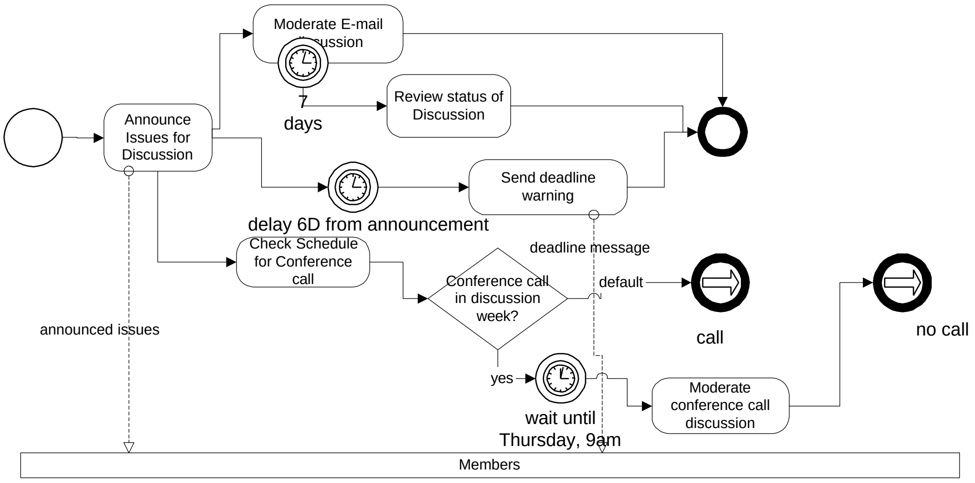

Moderate E-mail Discussion
7 days
Review status of Discussion
Announce Issues for Discussion
Send deadline warning
delay 6D from announcement
Check Schedule for Conference call
deadline message
Conference call in discussion week?
default
call
no call
announced issues
yes
wait until Thursday, 9am
Moderate conference call discussion
Members

Figure 5 : Example of the BPMN process

The notation also offers a symbol, which represents the data object. Primary purpose of modeling data objects is to increase the understanding of the process model and it does not have an influence on the automated process execution. The pool [Figure 6] symbol usually represents an organization. Sequence flow of the process cannot cross the pool boundaries. The pools can communicate with each other only using message flows. One pool can contain more swim lanes, which represent divisions or the responsible roles in the organization.

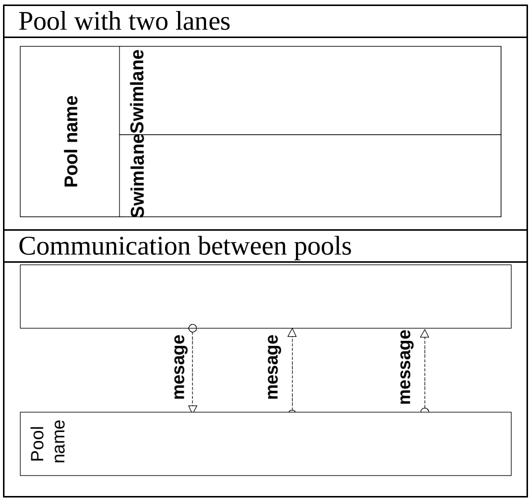


Figure 6 : Pools and swimlanes

Described graphical symbols doesn't represent the whole set of BPMN symbols, because the notation is extensible from the software tool vendors and the current version of the BPMN specification (0.9) is not complete.

V. MAPPING BPMN DIAGRAM TO EXECUTABLE FORM (BPML)

Until now, there wasn't a standard, commonly accepted and independent way to map graphical process model to executable form. BPMN contains mapping rules, which can be used to map its graphical symbols to XML tags, which are valid for two languages: BPML and BPEL4WS. The example of such mapping will be shown in the following paragraph.

[Figure 5] shows the part of the discussion moderation process for virtual team. The process begins with discussion issues announcement and takes a week to complete. The following graphical symbols were used: events (start, stop, timer, link), tasks, decisions, sequence flows, message flows and pools. The result of the mapping process is shown in [Figure 7].

```
<process name="DiscussionCycle">
<sequence>
<action name="AnnounceIssuesforDiscussion"
portType="tns:emailPort"
operation="sendDiscussionAnnouncement"/>
<all>
<call
process="ModerateEmailDiscussionProcess"/>
<sequence>
<delay name="Delay6daysfromAnnouncement"
duration="P6D"/>
<action
name="EMailDiscussionDeadlineWarning"
portType="tns:emailPort"
operation="sendDiscussionWarning"/>
</sequence>
<sequence>
<action
name="CheckCalendarforConferenceCall" … >
<output property="ConCall" element="…" />
</action>
<switch
name="ConferenceCallinDiscussionWeek">
<case name="Yes">
<condition>ConCall=true<condition/>
<delay name="WaituntilThursday9am"
dateTime="P6DT9H"/>
<action name="ModerateConferenceCall
Discussion" … />
<raise signal="NoCall"/>
</case>
<default name="Default">
<raise signal="Call"/>
</default>
</sequence>
</all>
</sequence>
</process>
<process
name="ModerateEmailDiscussionProcess">
<action name="ModerateEmailDiscussion" …
/>
<context>
<schedule code="OneWeek" duration="P7D"/>
<fault>
<case code="OneWeek">
<action name="ReviewStatusofDiscussion"
portType="tns:internalPort"
operation="receiveDiscussionStatus">
<output property="DiscussionOver"
element="…"/>
</action>
</case>
</fault>
</context>
</process>
```

Figure 7 : Mapping to the BPML

VI. DISCUSSION

Usage of the early version of BPMN showed itself as a good choice. It proved that notation is very expressive and easy to learn, which is mainly due to previous experience of the authors with process modeling in EPC and Petri nets notation. Although it is reasonable to wait with its usage until the final version and supportive tools are released.

When (If) this notation is commonly accepted and becomes also a de-facto standard, it can be predicted, the repositories of processes and process patterns will emerge. But, searching for the appropriate process can become burdensome, similar as difficulties when searching world wide web for documents. Not only searching, also choosing the right process for organization can be a difficult task, mainly because the area of process relations and their properties (e.g. complexity) is still in the development. One of the possibilities to improve classification of the process models is to describe them before publishing using some resource description language, for example RDF or DC (Dublin core classifiers). These descriptions could help to ones who are responsible for discovering, describing, modeling and deploying the process models. If the process models are described in such a way, we could search, for example, all the similar processes to our process, or, previous versions of the process.

[Figure 8] shows the example of the semantic web of the process models using DC classifiers, where Px represents part of the process model or whole process model and A represents a person.

For description of the process models, the following DC classifiers (tags) could be used:

- *creator* - Author of the process ()
- *relation.isVersionOf*, which could represent similarity or evolution of the process model
- *relation.Replaces|isReplacedBy*, with this classifier, the ‘no longer used’ process models can be tagged
- *relation.isPartOf|hasPart*, which could be used to describe the aggregation of the process part
- And others

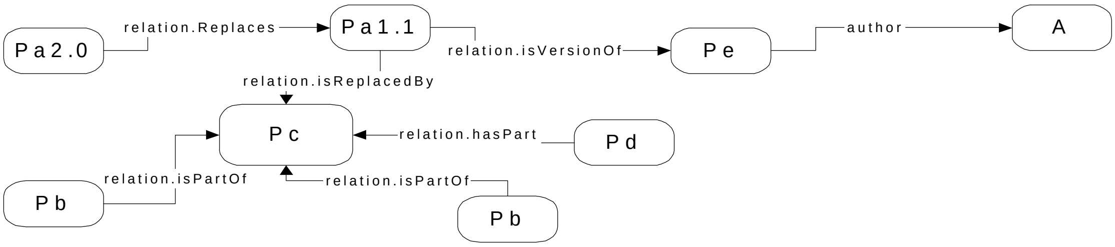


Figure 8 : Process models semantic web

VII. CONCLUSION

Question mark on the end of the title of the article has been written with a reason: why would this notation succeed to gain broader public attention as other process modeling techniques? The answer could be, BPMN technique is based on well-known and proven concepts, it is user-friendly and allows straightforward mapping to process execution languages. The BPMN specification is not owned by single corporation, but it is developed by many organizations (IBM, BEA, IDS Scheer, Fujitsu). Therefore, for wide public adaptation it needs only the critical mass of users.

VIII. LITERATURE

[1] ROZMAN, Tomislav. *Razvoj in funkcija orodja za ocenjevanje kompleksnosti programskih procesov : diplomska naloga univerzitetnega študijskega programa*, Maribor, 2001.

[2] VAJDE HORVAT, Romana, ROZMAN, Ivan, ROZMAN, Tomislav. *How to evaluate the complexity of software processes?*, MIPRO 2000

[3] BPMI.org, *Business Process Modeling Notation*, Working Draft (0.9) November 13, 2002, http://www.bpmi.org/

[4] Assaf Arkin, Intalio, Business Process Modeling Language, 2002, http://www.bpmi.org/

[5] BEA Systems, International Business Machines Corporation, Microsoft Corporation, Inc., *Business Process Execution Language for Web Services*, Version 1.0, http://www.ibm.com/developerworks/library/ws-bpel/

[6] Dublin Core Metadata Initiative, *Dublin Core Qualifiers*, http://dublincore.org/documents/dcmes-qualifiers/