Selecting a Process Modeling Language for Process Based Unification of Multiple Standards and Models

Zador Daniel Kelemen¹, Rob Kusters², Jos Trienekens², Katalin Balla³, ⁴
¹ThyssenKrupp Presta Hungary
Daniel.Kelemen@thyssenkrupp.com,
²Eindhoven University of Technology
R.J.Kusters@tue.nl, J.J.M.Trienekens@tue.nl,
³Budapest University of Technology and Economics
KBalla@iit.bme.hu,
⁴SQI – Hungarian Software Quality Consulting Institute
Balla.Katalin@sqi.hu

Abstract This report summarizes the selection process performed in order to ensure a modelling language of sufficient quality for describing processes based on multiple software quality models and standards. UML activity diagrams, EPCs and BPMNs are compared based on well-defined criteria: intelligibility, ability of expressing process elements and workflow patterns, software support, portability and widespread. Both the criteria and the rationale of the final selection can be used as a basis for selecting a process modeling language for other projects in the industry.

Keywords: quality approach, process based unification, multimodel software process improvement, BPMN, EPC, UML.

1 Introduction

The objective of this technical report is to present the selection of a process modeling language for Process Based Unification (PBU) of multiple quality approaches.

We call quality approach each quality model, standard (e.g. ISO 9001 or ISO 12207), method and (improvement) technology framework (e.g. CMMI, SPICE). By Process Based Unification we mean mapping quality approaches to processes [1], [2].

In order to have a sound basis for representing a unified process (the main result of Process Based Unification), we considered important studying modeling languages and selecting an appropriate one.

Many of the current research focuses on investigating process modeling languages (e.g. [3]–[5]). The practice indicates that graphical representation of processes helps understanding them, and can serve as a good support in understanding the (related) textual descriptions. Beside that a graphical
representation can be more easily interpretable than a multi-page text, it condenses information and can be applied in representing unambiguously the junctions, cycles and decision points in the process, enhancing the clarity of information.

Due to that sometimes it is really difficult to create unambiguous textual process descriptions, which are understandable for everyone, a graphical representation often is the best accessory of the process descriptor. A graphical representation helps solving problems of textual-only descriptions by eliminating the potential textual inconsistencies. Furthermore, it is easier to process, verify, validate and maintain the process representation.

A process descriptor best practice is to prepare the graphical representation first, then to attach a text explanation to the elements of the graphical representation, primarily to processes, steps, activities and inputs/outputs. Obviously, the process descriptor can decide the order of description and modeling, based on his/her situation and information available.

**Modeling methods and architectures**

Process modeling can be categorized from several points of view, e.g. according to modeling approach or modeling architecture. According to the minimalist approach, only the inevitably important elements are necessary to model. In this case, the modeler focuses on a representation which can be digested quickly and easily. In the maximalist approach the goal is to create models which can be digested by computers. Models created in a maximalist way may imply a number of details which are more difficultly understandable and less clear for people.

From the architectural point of view, two principal directions can be observed: the top-down and bottom-up approach. The top-down method starts from the idea of the process, first describing it, writing down an ideal process and then performing it. As opposed to this, the bottom-up method tries to model existing, working processes.

In this research our goal was to explore existing processes and to model only their important elements; thus we followed a bottom-up, minimalist approach. The comparison made in this report is based on this approach as well.

**Workflow patterns**

Workflow (or control flow) patterns show the expressiveness of a graphical representation language. Workflow patterns are e.g. the sequence, different types of junctions or synchronization.

At the selection of a modeling language an important viewpoint may be the way in which the most frequently used workflow patterns are represented. In one work at the Eindhoven University of Technology, 21 workflow patterns were identified and assigned to different categories [6]. Researchers examined how the selected patterns can be realized in BPMN and EPC. Another researcher, [7] on a similar manner to the former research, demonstrates how the workflow patterns can be represented in different process modeling languages. Having their results, one of our primary comparison aspects has to be the ability of the modeling language of expressing workflow patterns.
2 Approach

In selecting a process modeling language for Process Based Unification of multiple quality approaches we defined the following steps:
1. Selecting a subset of candidate process modeling languages (section 3),
2. Reviewing candidate process modelling languages (sections 3.1-3.3),
3. Defining a comparison criteria for selecting a process modeling languages (section 4),
4. Comparison of process modeling languages based on criteria (section 5),
5. Selecting a process modeling language (section 6).

3 Candidate Process modeling languages

In this section we present the information gathered while we were looking for the characteristics and features of process modeling languages.

In software developer community UML (Unified Modeling Language) activity diagrams are applied in many cases for process descriptions, but we may face several other methods, as: Petri nets, EPC (Event Driven Process Chain), Workflow networks, YAWL (Yet Another Workflow Language), GPWL (Graph-Based Workflow Language), BPMN (Business Process Modeling Notation) and BPD (Business Process Diagram). Besides these, less formalised solutions SPEM and POEM (Process Oriented Enterprise Modeling) should also be mentioned.

We analysed the structure and content of different quality approaches and we concluded that several elements can be converted to process elements [2], [8], [9]. Such convertible elements of process-based quality approaches are inputs, outputs, typical work products, tasks or activities. Based on an investigation we also defined a set of process elements which could provide the basis for Process Based Unification[1], [2].

Amongst the mentioned solutions BPMN, EPC and UML seems the most plausible for us, because:
- well-known workflow patterns can be easily used with them,
- they cover process elements identified,
- they have standardised and uniform components,
- they are suitable for human and machine processing equally,
- they are widespread,
- a wide-range software support is available for them.

A wide scientific and practical literature exists on process modeling methods, therefore only a brief overview will be presented here.

3.1 BPMN

BPMN (Business Process Modeling Notation) is a standardised graphical notation for representing business processes in workflow. BPMN was developed by BPMI (Business Process Management Initiative) and it is handled by OMG
(Object Management Group) since the fusion of the two organizations. The goal of BPMI with BPMN was to create a standardized notation which is understandable for all relevant stakeholders. Such stakeholders might be business analysts – who create and refine the processes, developers – who are responsible for the realisation of certain processes or managers – who manage and control the processes. Therefore, BPMN is a common graphical notation with the goal of eliminating the communication gaps amongst stakeholders with different background. It is important to emphasize that BPMN is built on flowchart technique; during its development several notation systems and methods were used, such as UML activity diagrams, UML EDOC Business Process, IDEF, ebXML BPSS, Activity-Decision Flow (ADF) diagram, RosettaNet, LOVeM and EPC [10].

3.1.1 Elements of BPMN

BPMN includes some well-defined basic elements, which can be ordered into the next categories:

- flow objects: event, activity and gateway;
- connecting objects: sequence flow, message flow, association;
- swimlanes: pool, lane;
- artifacts: data objects, group, annotation;

![Figure 1 – Basic symbols of BPMN, source: [11]](image)

3.1.2 Diagram: BPD

BPD-s (Business Process Diagram) can be created using BPMN notation. The following types of BPDs exist:

- high-level business,
- detailed business,
- cooperating with unknown external processes,
- collaboration BPD.
In a single model, different types of BPDs can be used. However, if too many types of BPDs are used simultaneously in one diagram, it can turn into one which can be understood quite difficultly. Therefore, it is advisable to use always only one type of BPD.

3.1.3 Storage format: WS-BPEL

The language WS-BPEL (Web Services Business Process Execution Language) or BPEL was created to describe executable business processes. The specification of BPMN includes a description about how the BPMN can be converted to BPEL codes, but unfortunately this mapping is informal and incomplete [12]. As BPMN and BPEL are becoming widespread, several tools are trying to implement the conversion between these two. An open-source example tool is BPMN2BPEL.

In the course of the development of such tools, fundamental differences between BPMN and BPEL came into light. Because of these differences, in a number of cases BPDs cannot be converted to BPEL. The situation may become more complicated when the goal is to generate BPEL codes readable by humans, or round-trip engineering is used during the development [13].

3.1.4 Storage format: XPDL, software support

XPDL (XML Process Definition Language) is a format standardised by the Workflow Management Coalition (WFMC). The standardized format provides the permeability amongst different workflow products, modelers and business process management tools. XPDL defines an XML scheme for describing the declarative part of business processes. In XPDL graphical and semantic information of the business processes can be stored. XPDL was designed so it can store all the information needed in BPDs, e.g. two dimensional coordinates, which show the position of BPMN elements in the diagram. It also contains information related to the execution. This is an important difference between BPEL and XPDL, because BPEL concentrates only on the executable aspects of BPDs. BPEL does not contain elements that would help in the graphical representation. For more details see the XPDL specification at [14].

3.2 EPC

The EPC method was developed within the frame of project ARIS by prof. Wilhelm-August Scheer at Saarlandes University at the beginning of 90’s. It was firstly used at SAP, but nowadays a number of companies are using it for modeling, analysing and redesigning their business processes. EPC is a directed graph of events and functions. Implies diverse connecting elements, with which the alternative and parallel execution routes can be described. It uses logical operators such OR, AND and XOR. The intelligibility and simplicity are one of its largest strengths. Unfortunately its syntax and semantics are not well-defined [15].
3.2.1 **Elements of EPC**

EPC elements are the following: event, function, process interface, connectors (AND, OR, XOR), control flow arc, participant (e.g. organization unit), application, data (information, material, and resource object) relation.

![EPC Diagram](image)

**Figure 2 – Symbols of EPC diagram, source: [16]**

3.2.2 **Storage format: EPML**

It is important to mention, that a standard, portable XML format was developed for storing EPCs, which is called EPML (EPC Markup Language). See [17] about the format, tools supporting the format and connected scientific and practical articles. A brief description of the format can be found in technical report [16].

3.2.3 **Software support**

The best-known software (10), which support EPC are the following: SAP R/3 (SAP AG), ARIS (IDS Scheer), LiveModel/Analyst (Intelllicorp Inc.), Visio (Visio Corp.), Visio (Microsoft), ADONIS (BOC Group), Semtalk (Sementation), Bonapart (Pikos), SmartDraw, EPC Tools (Paderborn Univ.).

3.3 **UML activity diagrams**

OMG defined 13 types of diagrams in UML 2.x, among which more are suitable for describing processes. Such types are e.g. sequence diagrams, activity diagrams, state-machine diagrams and communication diagrams [18]. From among these, the activity diagram is suitable and frequently used for describing processes.

3.3.1 **Elements of UML activity diagrams**

The list of elements of UML activity diagrams can be seen on the figure below. It includes the following main elements: state, transition, flow, decision, swimlane, signal receipt, signal send, constraint and note.
3.3.2 Storage format: XMI

Standard XMI (ISO/IEC 19503:2005 Information technology - XML Metadata Interchange) (ISO/IEC, 2005) was developed by OMG for describing XML metadata. It can be used to describe any kind of metadata, which can be expressed in MOF (Meta-Object Facility). Most frequent applications of XMI are the storage of UML models, but it is used to store models of other languages as well. Unfortunately, the XMI-based portability between UML modeling tools is still not solved because of incompatible XMI formats.

3.3.3 Software support

We could write much about software support of UML (activity diagrams), because it is a widespread modeling language on the software market. UML is supported by such software industry giants like IBM, Borland, Microsoft or Oracle. Besides them, several further commercial and freeware UML modeler products are available on the market. The list of tools supporting UML is expanding continuously; one list can be attained at [19].

It is difficult to make a selection or comparison without using criteria; therefore first we define the aspects which could be the most important at selecting a modeling language for representing the peer review process, and then we make the selection based on the criteria defined.

4 Comparison criteria

The primary criterion is the intelligibility, since models will be interpreted, used or modified by people coming from different environments. The intelligibility of a modeling language is a characteristic depending on an ability and background of the interpreter, thus it is difficultly measurable. Nevertheless some researchers address this topic, e.g. [20] or [21].

The ability to represent workflow patterns shows the expressiveness of the language. The complexity of models increases with a less expressive language. The intelligibility decreases and the number of mistakes grows by the increase of the complexity of models.

Besides the expressiveness of the language, coverage of process elements identified in [2] is required.

As we would like to create and modify our process models quickly and easily, another goal is to have a suitable software support for the chosen language.
The administrative organizations providing different services may use software which are independent from each other therefore, we consider important the selected approach not to be dependent on one single manufacturer. Besides freely choosing the software used for modeling, saving into a portable format may also be useful.

The more a technology or a modeling solution is used, the easier its introduction and acceptance. As a result, it is important to choose a solution as widely applied as possible.

Summarizing, we considered the following aspects:

- intelligibility,
- coverage of process elements,
- ability of expressing workflow patterns,
- software support,
- portable format,
- widespread in different areas.

These are the comparison criteria for selecting the process modeling language.

5 Comparison

In the course of comparison we value the approaches based on the criteria presented. We deal with the criteria in order of their importance. The order of importance was decided by us taking into account project needs.

5.1 Intelligibility

73 undergraduates and 12 process modeling experts were involved from the University of Vienna, Technical University of Eindhoven and the University of Madeira into a research performed by Mendling et al [20]. They asked participants to fill in questionnaires, then they analysed how the participants can interpret the 12 models presented. The research showed that understanding the models is primarily influenced by personal factors like the participants' previous theoretical and practical process modeling knowledge and experience. The other important factor in terms of the intelligibility is the model size, which can be measured with different metrics [20], [22]. Additional possible influencing factors of the intelligibility could be the expertise of the interpreter in the modeling language, the components of modeling language, the layout and arrangement of figures. In terms of intelligibility [6] we did not see any difference between UML 2.0 activity diagrams and BPDs.

From among the three modeling languages, software designers and developers use UML. EPC is mainly spread in the business sphere, while BPMN is present in both areas. This is not by chance; the goal of BPMI with the BPMN was to create a standard notation which is intelligible for people with different backgrounds.
Therefore, we consider BPMN the most appropriate approach from the first point of view.

5.2 Coverage of process elements

In order to model processes described using process elements needed for Process Based Unification, we need to select a language which can represent these process elements. In Table 1 we map process elements identified to the three preselected process modeling language elements. In order to understand how people perceive PML elements, besides the official specifications we looked at additional tutorial sites e.g. [23] or [24].

Table 1 shows that most of the process elements identified previously are present in BPMN, EPC and UML activity diagrams, therefore we consider them equally good for representing processes.

Table 1 – Coverage of process elements

<table>
<thead>
<tr>
<th>Process element</th>
<th>BPMN</th>
<th>EPC</th>
<th>UML Activity diagram</th>
</tr>
</thead>
<tbody>
<tr>
<td>Process name</td>
<td>Process name</td>
<td>Function name</td>
<td>Activity name</td>
</tr>
<tr>
<td>Process parent</td>
<td>Parent-child relation (unlimited levels)</td>
<td>–</td>
<td>Activity-Action relation</td>
</tr>
<tr>
<td>Process purpose</td>
<td>In textual descriptions</td>
<td>In textual descriptions</td>
<td>–</td>
</tr>
<tr>
<td>Activity</td>
<td>Activity</td>
<td>Function</td>
<td>Activity</td>
</tr>
<tr>
<td>Role, responsibility</td>
<td>Lanes</td>
<td>Organisation unit</td>
<td>Partition or Swimlane</td>
</tr>
<tr>
<td>Artifact</td>
<td>Artifact</td>
<td>Document, Information object</td>
<td>Object</td>
</tr>
<tr>
<td>Element relation</td>
<td>Element relations</td>
<td>Element relations</td>
<td>Control flow, Object flow</td>
</tr>
<tr>
<td>Entry/exit criteria</td>
<td>–</td>
<td>–</td>
<td>Action constraints (pre and postconditions)</td>
</tr>
</tbody>
</table>

5.3 Ability of expressing workflow patterns

All three modeling languages are built on similar elements. Such elements are the fork-join, branch-merge, basic and extended activities, but these are defined using different notations. The equivalent of some elements do not exist in the other diagrams, which make the automatic conversion difficult. Wienberg describes such notation differences between EPC and UML activity diagrams [25]. Making use of Aalst and other’s results, White compared BPMN and UML activity diagrams based on 21 fundamental workflow patterns [6]. As the result of the comparison, he came to the conclusion that both languages are equally
appropriate for describing the selected workflow patterns. He remarks one single exception, that the meta-model of activity diagram does not have a suitable structure to describe a certain workflow pattern. White concludes that the two approaches are different views of a single meta-model [6]. Since OMG handles both BPMN and UML activity diagrams, these might converge to each other in the future.

EPC was examined by Mendling et al, based on the 20 patterns of Aalst. Deficiencies were discovered and corrections were also proposed [26].

It is difficult to rank the three approaches from this point of view; all the three can represent the most of fundamental workflow patterns. However, a distinction is needed to be made: based on White’s results we consider BPMN the most appropriate.

5.4 Software support

Comparing their software support, it is visible that the most supported solution is the UML (activity diagram), BPMN comes after this, then EPC. The disadvantage of EPC’s software support is connected to its origin; it was developed and used at a single company, while UML and BPMN are company-independent, handled by OMG. Therefore, we consider UML the most software-supported solution. In spite of the fact that most modeler tools support UML, it is possible to find suitable solutions for the other two languages too. E.g. all three types of diagrams can be created with MS Visio.

5.5 Portable format

In terms of the portability EPC’s strength is EPML defined by Mendling and Nüttgens [16], which is a standard and independent XML format. The question is that, how the different applications will support it. In the case of UML diagrams, several UML modeler tools support the XMI format (defined by OMG, currently still with incompatibility issues) or can export to some other XML formats. In the case of BPMN, in order to support portability the Workflow Management Coalition defined the XPDL format.

It is visible that all three formats are based on XML and freely available, but deficiencies exist in all three cases on the side of implementation. The opportunity is given; the application of the standardised format depends on the developers of modeling tools. Formats supporting BPMN and UML were specified by a consortium. Therefore, they may count on a bigger support than EPC.

From the portability point of view, XMI format was implemented by several manufacturers therefore, we consider the UML the most portable modeling language.
5.6 Widespread in different areas

Due to the OMG's professional past, its standardized notations are accepted and adopted quickly by developer companies. We consider this a determining factor in the acceptance and adoption of UML and BPMN. EPC was introduced at a concrete company with no consortium behind it, which resulted in a more modest spreading compared to the former two. The IT sector quasi exclusively uses UML for modeling. In the business modeling multiple notations are used, starting from the simple flowcharts towards the BPMN, EPCs and UML; therefore it is difficult to tell which one is the most widespread in this area. Because of UML is spread in both areas, we consider it the best solution for our goal.

6 Selecting a process modeling language

Based on the comparison made, we can state that all three notations have a number of benefits; in addition, in the most cases the differences between them are minimal. At the same time, taking into account the previously defined criteria, BPMN seems the most appropriate solution from the intelligibility and expressiveness point of view. On the other hand, UML activity diagrams have the largest software support, best portable format and are widespread.

It is difficult to make the "best choice" among the presented modeling solutions without compromise. In spite of the fact that UML is adequate from more aspects than BPMN, we choose BPMN as a final solution, because:

- it is adequate from the first three most important aspects,
- a proper software support is available,
- very similar formats are available for all three languages in terms of portability,
- it is widespread in the business sphere; due to its intelligibility and OMG’s support its widespread can be growing quickly.

Table 2 summarises the result of comparison, listing the aspects according to their order of importance.

<table>
<thead>
<tr>
<th>Criterion</th>
<th>BPMN</th>
<th>EPC</th>
<th>UML activity diagram</th>
</tr>
</thead>
<tbody>
<tr>
<td>Intelligibility</td>
<td>+</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Coverage of process elements</td>
<td>+</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>Ability of expressing workflow patterns</td>
<td>+</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Software support</td>
<td></td>
<td>+</td>
<td></td>
</tr>
<tr>
<td>Portable format</td>
<td></td>
<td></td>
<td>+</td>
</tr>
<tr>
<td>Widespread</td>
<td></td>
<td></td>
<td>+</td>
</tr>
</tbody>
</table>

We analysed the structure and content of different quality approaches and we distinguished several quality approach elements. Such elements of process-based quality approaches are e.g. inputs, outputs, typical work products, tasks
or activities (see [2], [8], [9]). We mapped these quality approach elements to process elements. Based on the quality approach elements mapped to process elements and the comparison made above, we think BPMN can be a good solution for representing unified processes graphically.

7 Limitations

Selection criteria include the ability of expressing process elements. Other projects may require further process elements, or may require more formalism (e.g. for automation purposes more formal languages such as Petri nets may be considered). For that cases another language may be chosen.

We believe the criteria and the selection both were documented in a repeatable and reliable way.

8 Conclusion

In this report we discussed the selection of a graphical process representation format for Process Based Unification of multiple process oriented software quality approaches.

The comparison presented is related to a research having the goal of unifying models and standards by creating a unified process. For this unified process model, process elements were previously identified and the process modeling language was chosen accordingly.

In selecting a process modeling language for Process Based Unification candidate process modeling languages were presented and a subset (BPMN, EPC and UML activity diagram) was discussed in detail. Comparison criteria were also defined which consist of intelligibility, ability of expressing process elements and workflow patterns, software support, portability and widespread. The main criteria for our project were the intelligibility and the representation of a predefined set of process elements. For both of these criteria BPMN seemed an excellent solution and was chosen.

Both the criteria and the rationale of the final selection can be used as a basis for selecting a process modeling language for other projects in the industry.

---

1 This report has been published as a part of a PhD thesis [2] and it is part of a series of technical reports which include: [27]–[30].
References


