Business Process Modeling Languages for Information System Development

Hazem M. El-Bakry
Faculty of Computer Science & Information Systems, Mansoura University, EGYPT
E-mail: helbakry20@yahoo.com

Nikos Mastorakis
Technical University of Sofia, BULGARIA

Abstract:
It is known that different process models are used during various phases of the Business Process Modeling (BPM) life cycle. Many efforts have been introduced to overcome the disintegration between these process models to provide a standard process model for design and execution. However, this integration is still a problem that mazes the researches in this field till now. This paper discusses the problem of translation between process models, which known as the conceptual mismatch. Different perspectives are described considering the case of Business Process Modeling Notation (BPMN) vs. Business Process Execution Language (BPEL).

Keywords: Business Process Modeling, Conceptual Mismatch Problem, Business Process Modeling Notation, Business Process Execution Language

I. Introduction
BPM is very important for information system development. Furthermore, it plays a key role in modeling the organizational processes. In addition, it has its life cycle that appears in the business process development life cycle stages (design, implementation, test, evolution). Moreover, it has a number of business models that appear in its life cycle. Each one has its own function to capture the process meaning, and has its use in one of the business process development life cycle phases. Particularly, this concept appears its functionality in the design phase of business process development. In this phase, business process is documented from a business analyst perspective to capture the process’s requirements, then this documentation serve as input to technical analyst who responsible for developing the technical process models that are the executable work-flow specification models. Lastly, these specifications are evaluated and implemented as a part of the life cycle. BPM has various modeling languages. In this paper, we concentrate on two types of those languages BPMN and BPEL.

II. Business Process Modeling Notation
What is needed in a notation language is to be intuitive and to support the execution of the process. The Business Process Management Initiative (BPMI) understands this need and produces the first specification of BPMN (BPMN1.0) in May, 2004 [1-8].

A) BPMN Goals:-
The primary goal of the BPMN specification was to provide a notation that is readily understandable by business analyst, business technical, and business people who manage and monitor business processes. Another goal of BPMN is to support the automatic mapping from BPMN model to the BPEL model.

B) Business Process Diagram (BPD):-
BPMN provide graph-oriented models by defining a Business Process Diagram (BPD), which relies on flowcharting technique. The four categories of elements that made up the BPD are:-
Flow Objects: define the behavior of business process.
Connecting Objects: used to connect flow objects.
Swimlanes: used to group activities in separate categories for different responsibilities.
Artifacts: used to provide additional information about the process.

C) BPMN vs. UML:-
Some can thought that BPMN is a modeling language like any other modeling languages, such as UML, its purpose is to output a specification or a notation about a process. However, this is not true as BPMN is a modeling
language differ from others languages. For example, BPMN and UML, these two modeling languages have some differences between them as listed in the Table 1.

## III. Business Process Execution Language (BPEL)

BPEL for web services defines a notation for specify a process behavior based on web services. Version 1.1 of BPEL was released in 2003 by OASIS. BPEL is used to model the behavior of process such as:-

- Sequences of process activities.
- Correlation messages and process instances.
- Recovery behavior.
- Relationships between process roles.

BPEL provide a block-oriented model that made up of elements as:-

- **Partner links**: used to provide a communication channel to a remote web service to be used in the BPEL process.
- **Variables**: are used to store message data of a web service and control information of a process.
- **Correlations**.
- **Basic activities**: the basic activities such as, send message, reply, wait for response.
- **Structured activities**: such as, parallel execution and conditional branching based on data.
- **Handlers**: to deal with unexpected situations.

An example of how to transform BPMN model to a web service using BPEL is shown below (Table 2 & Figure 2)

## IV. Conceptual Mismatch Between Modeling Languages

The business process life cycle have different stages as we said before, modeling languages could be applied to various stages of the life cycle, each with different focus on modeling purpose. Some languages provide a high-level of specification that is required to understand the organization from a social perspective through a set of conceptual models (such as BPMN models). Other languages provide a low-level specification that is required to determine how to execute the specification by a work-flow engine through technical models (such as BPEL models).

Translation between these models is very helpful for both business analysts and technical analysts to make a well process specification and implementation, but while not all BPMN models has a corresponding BPEL models this type of translation produce a problem of loss of process semantic or what is known as the conceptual mismatch between process modeling languages deployed in different phases of the BPM life cycle.

When deriving a translation between process models, there are different BPM life cycle perspective must be taken into account:-

- **Domain representation capability mismatch**: - From business analyst perspective, the translation between BPMN and BPEL must preserve semantic information, that is, it should avoid loss in semantic representation information.
- **Control flow support mismatch**: - From a technical analyst perspective, the underlying work-flow execution engine determines the specification of a process, so that the control flow of a process is a central aspect that must be supported by the work-flow execution engine.
- **Process representation paradigm mismatch**: - From the process representation paradigm perspective, representation paradigm underlying any business process modeling language. The most common representation paradigms are graph-oriented paradigm (used in BPMN) and block-oriented paradigm (used in BPEL), there is no adopt that there is a conceptual mismatch between those different paradigms.

## V. Conclusions

In this paper, the problem of conceptual mismatch between process modeling languages has been introduced. Due to their importance in this field, the conceptual mismatch between BPMN and BPEL modeling languages has been discussed. It has been presented that BPMN is a powerful modeling language. This is because it provides a process model that is easily readable by business stakeholders, and also supports mapping to execution models. BPEL is a modeling language that supports the creation of a web service corresponds to a process. The problem is that mapping between BPMN and BPEL is not simple and can not be applied to all
cases of BPMN or BPEL because some of semantic information about the process is lost. This is known as the conceptual mismatch between business process modeling languages. In order to solve this problem, this conceptual mismatch must be defined in order to identify and then overcome it. It can be concluded that conceptual mismatch can be identified from three perspectives. First, domain representation capability mismatch, in which the selected domain representation must avoid loss in the semantic information of a process. Second, control flow support mismatch which implies that the control flow of a process is a central aspect and must be supported by the workflow execution engine. Third, process representation paradigm mismatch at which different representation paradigms require different transformation strategies.

References


Figure1: Example of a BPMN model [1]
### Table 1: BPMN vs. BPEL

<table>
<thead>
<tr>
<th>BPMN</th>
<th>UML</th>
</tr>
</thead>
<tbody>
<tr>
<td>BPMN is similar to UML behavior diagrams, but each of the two languages has very different approaches for modeling business processes.</td>
<td>UML is the result of the collective best modeling methods so; diagrams are not designed to work with each other.</td>
</tr>
<tr>
<td>BPMN provides a process-centric approach, makes it most appropriate to business analysts.</td>
<td>UML provides an object-oriented approach, makes it alien to business analysts.</td>
</tr>
<tr>
<td>BPMN enable the generation of the executable models BPEL4WS.</td>
<td>Doesn’t define any executable models with it, any execution model must be defined using Model Driven Architecture (MDA).</td>
</tr>
</tbody>
</table>

### Table 2: Mapping web service properties into partner link [8]

<table>
<thead>
<tr>
<th>BPMN Object/Attribute</th>
<th>BPEL Element/Attribute</th>
</tr>
</thead>
<tbody>
<tr>
<td>Implementation = &quot;Web service&quot;</td>
<td>Invoke, but some properties, below, will map to a partnerLink</td>
</tr>
<tr>
<td>Participant = &quot;ProcessStarter&quot;</td>
<td>name=&quot;ProcessStarter&quot; partnerLinkType=&quot;ProcessStarterPLT&quot;</td>
</tr>
<tr>
<td>BusinessRole = &quot;TravelProcessRole&quot;</td>
<td>myRole=&quot;TravelProcessRole&quot;</td>
</tr>
<tr>
<td>Participant = &quot;HotelReservationService&quot;</td>
<td>name=&quot;HotelReservationService&quot; partnerLinkType=&quot;HotelReservationServicePLT&quot;</td>
</tr>
<tr>
<td>BusinessRole = &quot;HotelReservationRole&quot;</td>
<td>myRole=&quot;HotelReservationRole&quot;</td>
</tr>
</tbody>
</table>

### Figure 2: Setting up the partner link elements for the process [8]

```xml
<partnerLinks>
  <partnerLink myRole="travelProcessRole" name="ProcessStarter" partnerLinkType="wsdl5:travelProcess"/>
  <partnerLink name="HotelReservationService" partnerLinkType="wsdl5:HotelReservationPartnerPLT" partnerRole="HotelReservationRole"/>
  <!-- Another 3 partnerLinks are defined -->
</partnerLinks>
```