Scientific Journal of Riga Technical University
Computer Science. Applied Computer Systems
2010
Volume 43

Towards the Mapping of Multidimensional BPMN Models to Process Definition Standards

Ludmila Penicina, *Riga Technical University*

Abstract **- Business Process Management (BPM) community has accepted Business Process Modeling Notation (BPMN) as de facto business process modeling standard. In this research BPMN is chosen as the notation for representation of multidimensional business processes, as it uses the set of graphical elements that can be refined with specific attributes, allowing to describe every important aspect of the business process, which can be regarded as the modeling dimension. However, BPMN is only graphical notation and does not support the essence of BPM – full business process lifecycle support: from documenting and modeling to execution and monitoring. To accelerate the adoption of BPM, industry has come forward with two BPMN complementing standards – Business Process Execution Language (BPEL) and XML Process Definition Language (XPDL). These two standards are used for different purposes – BPEL is process execution language and XPDL is process exchange language. The problem of mapping BPMN models to BPEL and XPDL is covered in various sources. This paper addresses the issue of mapping BPMN process models in different dimensions to BPEL and XPDL and explores how the structure of BPEL and XPDL is affected after the process is transformed to another dimension. The findings during experiments accomplished in this research can contribute to the creation of the guidelines of how BPMN model in different dimensions can be executed using BPEL or interchanged between different modelling tools.**

Keywords: **BPMN, BPEL, business process dimension, business process modeling, XPDL**

I. Introduction

The representation of work in the form of business process models can be considered as a method for enterprise systems analysis [1]. In the analysis phase of a software system project business process models encourage the communication between business and IT representatives. This communication helps to ensure that business process models capture business and software requirements identified by domain analysts and accepted by stakeholders. These models become an input for the development phase of the project. Business Process Modeling Notation (further in text - BPMN) provides a graphical notation for business process modeling. It defines a Business Process Diagram (BPD) [2], which is a kind of flowchart made up of elements for modeling business logic (e.g. business rules, event-based choices, etc) [1].

However, there exist two limitations of pure BPMN models that must be considered:

1. BPMN models are just description of the work done in the enterprise in the form of nice picture [1] – they are non executable;
2. BPMN defines only the look of how the process definition is displayed on the screen [3] – it does not define interchangeable storage format of the model.

To resolve the first issue – to make BPMN model executable – the development team has to create executable process definition or code from the given business process model by refining it with implementation details, so that it could be deployed to the process server. The BPM community has accepted Business Process Execution Language (further in text BPEL) as the standard for executable process definition [4]. BPEL is an XML based programming language enabling users to create formal descriptions of business process and business interactions in the form of Web services [5]. BPMN and BPEL can be used without each other, but BPMN model with no translation to BPEL is just a nice looking picture that to system developers makes no sense, but BPEL cannot be used for process analysis in the early phases of projects, because it is difficult for business representatives to read and understand it. The best case, in which BPM idea comes true, is when BPMN and BPEL are used in tandem. The scenario for such relationship is as follows. BPMN models are translated to BPEL code, so BPEL can be seen as a refinement for BPMN with implementation details such as data manipulation, Web service bindings, and other details [1].Using this scenario the BPMN model becomes a valuable input for development and the main task for the developers of process-oriented systems is to get executable code from BPMN.

To resolve the second issue – to make BPMN model interchangeable with other tools and to have a formal storage format for BPMN models – Workflow Management Coalition (WfMC) [3] has put forward the XPDL standard. XPDL goal is to store and exchange the process diagram. XPDL provides a file format that supports every aspect of the BPMN process definition notation including graphical descriptions of the diagram [3]. It is a file format that represents the “drawing” of the process definition. It has X & Y coordinates and node size. It has a concept of lines, and points along the line that give it a particular path. The nodes and lines have attributes, which can specify executable information such as roles, activity descriptions, timers, web service calls, etc [6]. It allows one process design tool to write out the diagram, and another to read the diagram.

As BPMN models become the foundation of the development and, as initially created model can be interchanged between parties using different tools, the following question arises – how to create high quality business process model that can be used through the whole business process lifecycle? In this paper high quality business process model means the model reveals all the recognizable details of the process flow. Knowing and understanding the details of business processes is important, because this gives the

opportunity to identify the bottlenecks and optimize business processes [8]. In order to reveal all the details of the business process it is necessary to analyze the process from different points of views and within different contexts, i.e., from different dimensions [7]. It means that the process that is represented, e.g., from performer's dimension is hiding in its structure the details about the process from the time dimension. In the result business analyst creates several models – one model of the same process for each dimension. In this paper BPMN is used for representing business process in different dimensions. BPMN swimlanes are used to organize and categorize activities of the process [2]. The idea of representing the process from different dimensions is based on [8]. This paper addresses the question how XML based process definitions (BPEL and XPDL) of the same business process change according to the change of the dimension of process representation?

The paper is structured as follows. The overview of related works is presented in Section II. The example of BPEL and XPDL process definitions generated from BPMN process models in different dimensions is shown in Section III. Conclusions and future work are described in Section IV.

II. Related Works

This section describes the basic challenges and the restrictions of mapping BPMN business process model to the executable BPEL process definition and XPDL, as well as introduces the idea of multidimensional business process modeling using BPMN.

A. BPMN Mapping to BPEL

BPMN 1.2 specification [2] provides a non-normative mapping from BPMN to BPEL, but the BPMN specification itself is known to be incomplete with respect to capturing all the required information for BPEL [2]. BPMN specification states that there are known issues with the mapping and the fixes to these issues will be incorporated in a later revision of the specification.

There exist numerous tools that support the execution of BPEL processes [9] – [11]. Some of the tools provide even the graphical editors for visualizing processes, however these editors follow the syntax of BPEL, not BPMN [9], [10]. The reason for that is that BPEL as any programming language defines far more syntactic restrictions than BPMN [1]. For example [1] defines that source BPMN business process has to contain no deadlocks or livelocks. The conclusion is that it is possible to represent processes in BPMN that cannot be mapped to BPEL. The BPMN mapping to BPEL is a challenge not only because of basis of these two standards – BPMN is a graph based notation, BPEL is a structured language based on blocks, but also because these two standards are maintained by different groups – Object Management Group (OMG) [12] is responsible for BPMN, BPEL was created by Advancing Open Standards for the Information Society (OASIS) [13].

B. BPMN Mapping to XPDL

XPDL 2.1 specification [3] provides documentation how BPMN can be mapped to XPDL. Structurally, BPMN and XPDL are very similar – both being flow-chart structures. In fact, the mapping between BPMN and XPDL is much more straightforward than the mapping between BPMN and BPEL. The XPDL provides design interchange because it offers one for one representation of the original BPMN process diagram – it can be written, and re-read to recover the original diagram. BPEL, on the other hand is a non- trivial mapping, which is widely recognized as being one-directional – we can take a BPMN diagram and produce BPEL, but it is difficult or impossible to recover the original BPMN diagram from the BPEL [6]. This is not surprising since BPEL was not designed for process design interchange. Every core elements of BPMN have a correspondence with a specific element of XPDL specifications.

XPDL has over 50 implementations and it is important to note that, XPDL provides vendor and user extension capabilities, so that it is possible to represent new graphic elements by extended attributes, embedded in XPDL specification.

C. Multidimensional Business Process Modeling in BPMN

A business process model is always modeled from a certain perspective of observation – e.g., time, business goals, performers, information and other perspectives [7]. Each perspective of observation may be regarded as a modeling dimension [7] or a combination of modeling dimensions.

When observing business process model from certain dimension, the observer is getting all the details only from represented point of view – process activities are organized according to the certain dimension, however business process analyst has to view the process from different dimensions to discover bottlenecks and areas of potential improvement in a process, the most time consuming process activities and the process nodes that could be optimized. In business process modeling phase it is essential for process model to reveal the following basic information about each of the activity performed in the business process:

1. Roles responsible for carrying out each activity in the process;
2. Start time/end time of the process and processing time of each activity in the process;
3. Documents exchanged within the process (inputs and outputs of each activity);
4. Business rules that control the workflow.

Each of above mentioned slots of information can be regarded as business process modeling dimension. According to each of these dimensions a process model can be created and then transformed to another dimension without losing process semantics, using BPMN syntax and revealing information about the process in new context. This paper covers only transformation between performer and time dimension. BPMN provides four basic categories of elements [2]:

1. Flow objects – events, activities, gateways;
2. Connecting objects – sequence flow, message flow, association;
3. Swimlanes – pools, lanes;
4. Artifacts – data object, group, annotation.

In [8] BPMN is proposed as official multidimensional modeling standard, because BPMN specification [2] describes a pool as the container for the sequence flow between activities, so a pool or a lane can be considered as a container for the flow of activities according to a particular dimension. How business process dimensions correlate with BPMN basic elements can be found in [8].

III. Generation of XML Based Process Definitions

This paper focuses on generating BPEL and XPDL from BPMN private business process, i.e., the process with no representation of business interactions. The future work will cover the analysis of BPEL and XPDL generation for B2B situations (collaboration processes), when business process model contains several executable pools and the message flow can cross the boundary of the pool.

This section presents examples of two BPMN business process models – the testing process in a software company and the simplified version of the payment submission process in the university. Both processes are modeled from performer and time dimensions, then BPEL and XPDL instructions were generated for each model.

In the case of the testing process the transformation to another dimension was quite simple and required only regrouping of the elements; therefore the BPEL and XPDL code generated from both models is almost equal only differing by the names of the swimlanes. During the transformation of the payment submission process to the time dimension three parallel gateways were removed. These elements were necessary in the performer dimension of the process because of BPMN syntactic restriction for modeling parallel activities and synchronization of these activities and were found to be irrelevant in the time dimension model, where all activities are sequential. These updates of the process model in the time dimension resulted in the changes of BPEL and XPDL process definitions.

A. The Testing Process

The business process presented in this section defines the order of work for a testing team in the software development company. There are two roles in the testing team – the testing manager and the tester, each participant is responsible for carrying out a particular set of activities.

Fig. 1 describes the testing process from the performer dimension – all process activities are grouped in the lanes by performer, lanes are organized in one common pool – Testing team. The process model shown in Fig. 1 does not reveal any information about time factor in the process – it is not defined when process is initiated and how long does it take for each activity to be executed [14].

Listing 1 shows the fragments of BPEL code describing the activities of BPMN model in the performer dimension [14]. Listing 2 shows small fragment of XPDL code generated from BPMN model in performer dimension.

The process model in the Fig. 2 reveals important information about the time factor in the process: process activities are organized in the lanes according to time dimension. This model shows how much work is done in the particular period of time [14].

After the source business process model is transformed to time dimension it still contains one executable pool called “Testing team”, however organization of the process activities in the pool differ from the initial model. But how transformation to time dimension changes the BPEL and XPDL code? In this case the semantics of the BPEL code after transformation to another dimension is not affected. The only thing that has been changed – is the names lanes since they have been changed by the modeler to show the granularity of the time dimension. In the case of BPEL the only thing that is different is the name of the swimlane in the section partnerLink. BPEL section partnerLink that is affected after the transformation is marked in Listing 1 with another colour. In the case of XPDL the contents of the section NodesGraphicsInfo differ from BPMN model in time and in performer dimension. XPDL section NodesGraphicsInfo that is affected after the transformation is marked in Listing 2 with another colour.

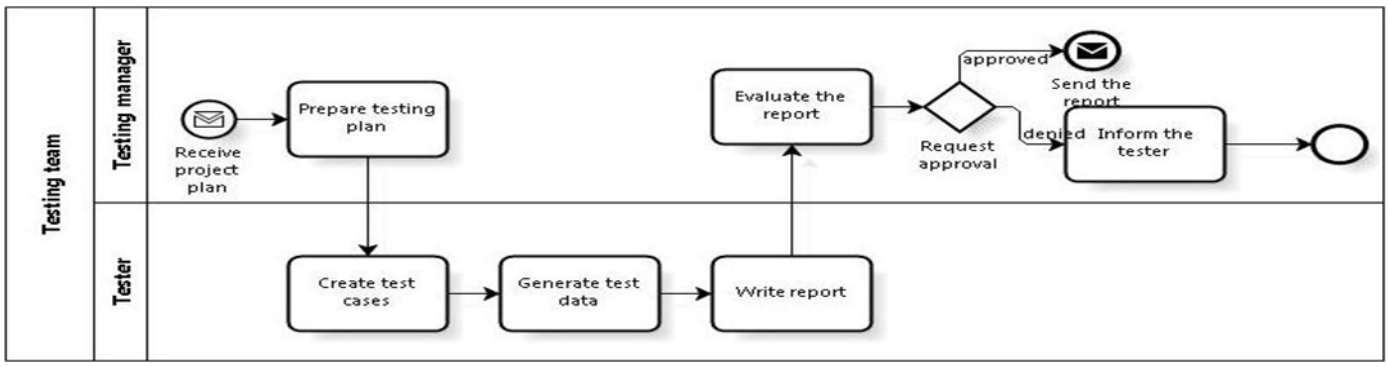

Fig. 1. Business process according to the performer dimension.

```
<bpel:sequence>
<bpel:receive
partnerLink="interfaceAndTESTING_TEAMPlkVar"
portType="this:Forinterface"
operation="Receive_project_plan"
variable="thisReceive_project_planRequestMsg
" createInstance="yes"
bpmn:label="Receive project plan"
bpmn:id="_9hooGgXEd65mJjXNiYS5g"></
bpel:receive>
<bpel:empty bpmn:label="Prepare testing
plan"
bpmn:id="_Kj2jgGgYEd65mJjXNiYS5g"/>
<bpel:empty bpmn:label="Create test cases"
bpmn:id="_DRgFUGgaEd65mJjXNiYS5g"/>
<bpel:empty bpmn:label="Generate test data"
bpmn:id="_kqBwQGgaEd65mJjXNiYS5g"/>
<bpel:empty bpmn:label="Write report"
bpmn:id="_8ZGfkGgYEd65mJjXNiYS5g"/>
<bpel:empty bpmn:label="Evaluate the report"
bpmn:id="_lNthkGgZEd65mJjXNiYS5g"/>
<bpel:if>
<bpel:condition>$thisReceive_project_planRes
ponseMsg.body</bpel:condition>
<bpel:sequence>
<bpel:empty bpmn:label="Inform the tester"
bpmn:id="_Yh4SsGgaEd65mJjXNiYS5g"/>
<bpel:empty bpmn:label="EventEndEmpty"
bpmn:id="_btsucGgaEd65mJjXNiYS5g"/>
</bpel:sequence>
<bpel:else>
<bpel:sequence>
<bpel:empty bpmn:label="Send the report"
bpmn:id="_Yx5WkGgYEd65mJjXNiYS5g"/>
</bpel:sequence>
</bpel:else>
</bpel:if>
</bpel:sequence>
```

Listing 1. BPEL code from BPMN in performer dimension

```
<Activities>
          <Activity Id="6c75adcc-3c37-4a23-
9b7a-c2c1ec356975" Name="Receive project
plan">  <Description />
            <Event>
              <StartEvent Trigger="Message">
            <NodeGraphicsInfos>
              <NodeGraphicsInfo
ToolId="BizAgi_Process_Modeler" Height="30"
Width="30"BorderColor="-10311914"
FillColor="-
1638505"><CoordinatesXCoordinate="137"
YCoordinate="120" />
              </NodeGraphicsInfo>
            </NodeGraphicsInfos>
            <Activity Id="cba4d0e9-436d-
43b2-987a-64f5af5da249" Name="Prepare
testing plan">
            <NodeGraphicsInfos>
              <NodeGraphicsInfo
ToolId="BizAgi_Process_Modeler" Height="60"
Width="90" BorderColor="-16553830"
FillColor="-1249281"><Coordinates
XCoordinate="220" YCoordinate="105" />
              </NodeGraphicsInfo>
            </NodeGraphicsInfos>
          </Activity>
          <Activity Id="c7f25d56-f4f5-43c6-
9e7a-5fa79c52e4db" Name="Create test cases">
            <NodeGraphicsInfos>
              <NodeGraphicsInfo
ToolId="BizAgi_Process_Modeler" Height="60"
Width="90" BorderColor="-16553830"
FillColor="-
1249281"><CoordinatesXCoordinate="380"
YCoordinate="105" />
              </NodeGraphicsInfo>
            </NodeGraphicsInfos>
          </Activity>
```

Listing 2. XPDL code fragment generated from BPMN in the performer dimension

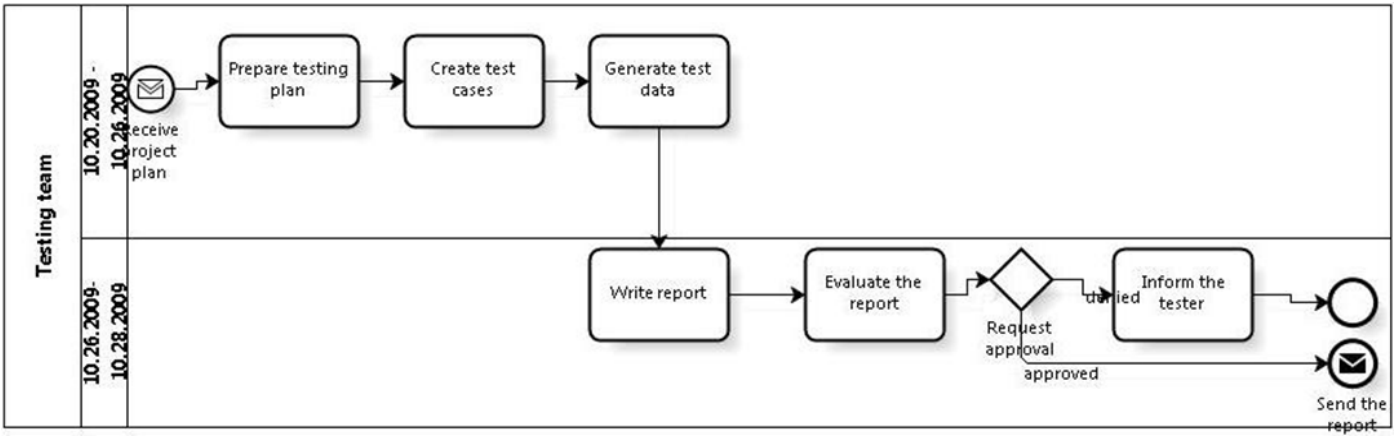

Fig. 2. Business process according to the time dimension

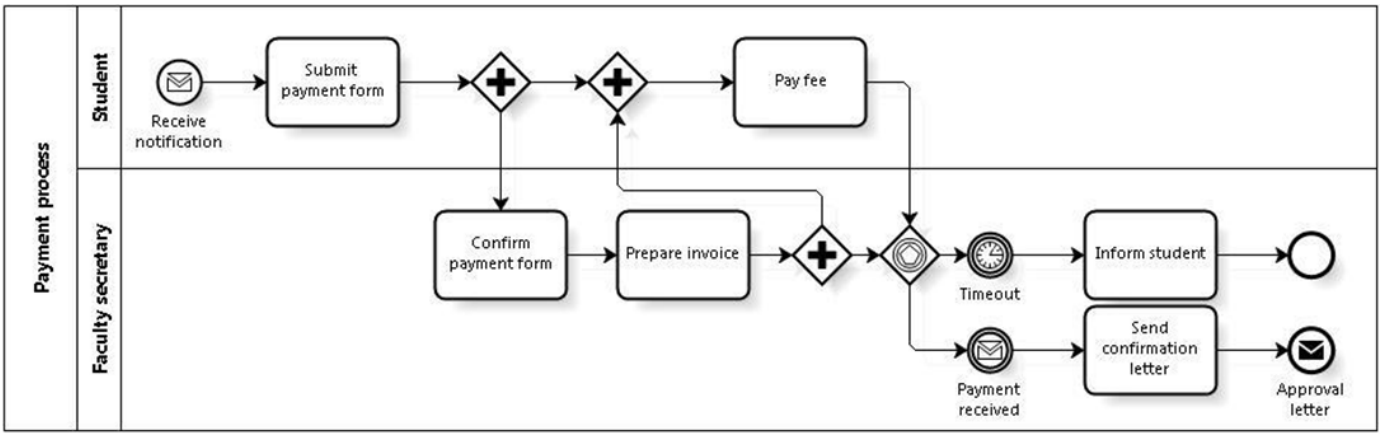

Fig. 3. Payment submission process model according to the performer dimension

B. The Payment Submission Process

The business process presented in this section defines what a student and the secretary of the faculty must do in order to process a payment to the university. There are two process performers – a student and the faculty secretary. The process model contains one executable pool named "Payment process" and two lanes – one for each performer [14].

The business process model shown in Fig. 3 contains 3 parallel gateways to provide a correct definition of parallel activities and synchronization construct that ensures that the student can pay only when invoice is ready and form is submitted. The model contains exclusive event-based gateway that assesses the different events the process might receive – the approval of the payment or the timeout – the first one to be received determines which outgoing sequence flow should be used [2].

Listing 3 shows fragment of BPEL code generated from BPMN payment submission process in performer dimension. BPMN element *parallel gateway* is mapped to BPEL construct *flow*, element *exclusive event-based gateway* is mapped to BPEL construct *pick* and BPEL constructs *onMessage* and *onAlarm* represent events the *exclusive event-based gateway* might receive [14].

Listing 4 shows fragment of XPDL code generated from BPMN payment submission process in performer dimension, that is responsible for handling the BPMN element *parallel gateway*. In XPDL *parallel gateway* is represented via construct *Route GatewayType="AND"* in the section *Activity*.

The business process model shown in Fig. 4 represents the payment submission process according to the time dimension. Its main difference from the model in performer dimension is that all activities are sequential and synchronization construct is unnecessary; during the transformation the parallel gateways were removed from model [14].

The BPEL code generated from model according to the time dimension does not contain the section <*bpel:flow*> (see Listing 5).

While BPEL code of the same BPMN model in performer dimension defines the same activities in the same section using the flow construct – for further details see Listing 3, the affected sections are marked with another colour.

```
<bpel:sequence>
<bpel:empty bpmn:label="Submit payment
form"/>
<bpel:flow bpmn:label="GatewayParallel"/>
<bpel:sequence>
<bpel:empty bpmn:label="Pay fee"/>
</bpel:sequence>
<bpel:sequence>
<bpel:empty bpmn:label=" Confirm payment
form"/>
<bpel:empty bpmn:label="Prepare invoice"/>
<bpel:flow bpmn:label="GatewayParallel"/>
<bpel:sequence>
<bpel:empty bpmn:label="Pay fee"/>
</bpel:sequence>
<bpel:empty/>
</bpel:flow>
</bpel:sequence>
</bpel:flow>
<bpel:pickbpmn:label="GatewayEventBasedExclu
sive">
<bpel:onAlarmbpmn:label="EventIntermediateTi
mer"><bpel:for>$thisEventStartMessageRequest
.body</bpel:for>
<bpel:sequence>
<bpel:empty bpmn:label="Inform student"/>
<bpel:empty bpmn:label="EventEndEmpty"/>
</bpel:sequence>
</bpel:onAlarm>
<bpel:onMessagebpmn:label="EventIntermediate
Message">
<bpel:sequence>
<bpel:emptybpmn:label="Send confirmation
letter"/>
<bpel:empty bpmn:label="EventEndMessage"/>
</bpel:sequence>
</bpel:onMessage>
</bpel:pick>
</bpel:sequence>
</bpel:process>
```

Listing 3. BPEL code from BPMN payment submission process in performer dimension

XPDL code does not contain 3 *Activity* sections that defined *parallel gateways* – see Listing 4, the affected sections are marked with another colour.

```
<Activities>
        <Activity Name="Receive
notification">
          <Event>
            <StartEvent Trigger="Message">
              <TriggerResultMessage />
            </StartEvent>
          </Event>
          <NodeGraphicsInfos>
            <NodeGraphicsInfo Height="30"
Width="30" BorderColor="-10311914"
FillColor="-
1638505"><CoordinatesXCoordinate="125"
YCoordinate="57" />
            </NodeGraphicsInfo>
          </NodeGraphicsInfos>
        </Activity>
        <Activity Name="Submit  payment
form">
          <NodeGraphicsInfos>
            <NodeGraphicsInfo Height="60"
Width="90" BorderColor="-16553830"
FillColor="-1249281"><Coordinates
XCoordinate="200" YCoordinate="42" />
            </NodeGraphicsInfo>
          </NodeGraphicsInfos>
        </Activity>
        <Activity>
          <Description />
          <Route GatewayType="AND" />
          <Documentation />
          <ExtendedAttributes />
          <NodeGraphicsInfos>
            <NodeGraphicsInfo Height="40"
Width="40" BorderColor="-5855715"
FillColor="-52"><Coordinates
XCoordinate="340" YCoordinate="52" />
            </NodeGraphicsInfo>
          </NodeGraphicsInfos>
        </Activity>
        <Activity >
          <Description />
          <Route GatewayType="AND" />
          <Documentation />
          <ExtendedAttributes />
          <NodeGraphicsInfos>
            <NodeGraphicsInfo Height="40"
Width="40" BorderColor="-5855715"
FillColor="-52"><Coordinates
XCoordinate="420" YCoordinate="52" />
            </NodeGraphicsInfo>
          </NodeGraphicsInfos>
        </Activity>
<Activity>
          <Description />
          <Route GatewayType="AND" />
          <Documentation />
          <ExtendedAttributes />
          <NodeGraphicsInfos>
            <NodeGraphicsInfo Height="40"
Width="40" BorderColor="-5855715"
FillColor="-52"><Coordinates
XCoordinate="560" YCoordinate="170" />
            </NodeGraphicsInfo>
          </NodeGraphicsInfos>
<IsForCompensationSpecified>false</IsForComp
ensationSpecified>
        </Activity>
```

Listing 4: XPDL code from BPMN payment submission process in performer dimension

```
<bpel:flow>:
<bpel:empty bpmn:label="Submit payment form"
<bpel:empty bpmn:label=" Confirm payment
form"
<bpel:empty bpmn:label="Prepare invoice"
<bpel:emptybpmn:label="Pay fee".
```

Listing 5: BPEL code fragment from BPMN payment submission process in time dimension

BPMN process definitions (both BPEL and XPDL) generated from time dimension are more readable as it contains less specific constructs due to removal of the synchronization construct and parallel gateways from BPMN model in time dimension. Generated instructions from time dimension model can be understood even by business representatives; however they define the same process, which is presented in Fig. 3. The further research must investigate whether synchronization constructs and parallel gateways are always unnecessary in business process model presented in time dimension.

IV. Conclusions and Future Work

This paper addresses the issue of generating XML based BPMN process definitions from BPMN private business processes presented in performer and time dimension. Business processes discussed in this paper describe the testing process in the software company and the payment submission process in the university. For each business process two BPMN models were created – the performer and time dimension model. The source model of each process was presented in performer dimension and then manually transformed to time dimension according to the guidelines presented in [8], however before transformation BPEL and XPDL instructions were generated from the initial model. Then BPEL and XPDL code was repeatedly generated from the model in time dimension.

The question that paper addresses is how XML based BPMN process definitions of the same process is affected after process is presented in another dimension? To answer the question the comparison of two BPEL and XPDL process definitions of each process was carried out – analyzing each line of two BPEL and two XPDL process descriptions.

In the case of the testing process it was concluded that BPEL code obtained from both models was the same, as well as XPDL, only the names of the swimlanes differed. The explanation for this finding is simple – in the case of the testing process the transformation to another dimension meant only regrouping the elements to different swimlanes, keeping all the relationships between the elements the same. The transformation was so trivial because the model did not contained such specific flow control patterns as merging of the flow threads or parallel activities.

The case of payment submission process model transformation was more complicated. It was revealed that in the model according to time dimension synchronization construct and parallel split construct are unnecessary, because all activities from time dimension are sequential. Due to this

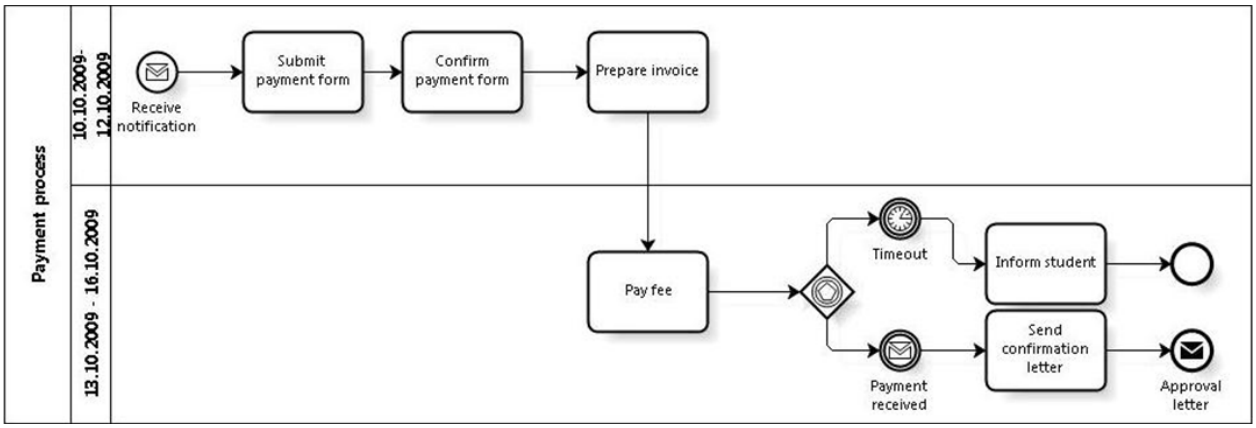

Fig. 4. Payment submission process model according to the time dimension

conclusion 3 parallel gateways were removed from the initial payment submission process. This resulted in the changes of the BPEL and XPDL instructions – in the BPEL code from time dimension model section *flow* is absent and the sequence of the activities is different, in the XPDL process definition sections Activity that defined *parallel gateways* using construct *Route GatewayType="AND"* was removed.

The listings of code presented in this paper clearly shows that nevertheless XPDL and BPEL are both definitions of BPMN process, they are used for entirely different purposes – XPDL is used to describe the graphical context of the BPMN diagram, but BPEL describes the executable context of the diagram.

The future work will address the question about what are the other specific workflow patterns (such as Discriminator, Multiple Merge, Multiple Choice, N out of M Join, Synchronizing Merge) that may be changed or removed during the transformation to time dimension.

The future research will use conclusions presented in this paper to create requirements for the modeling tool that is capable of creating BPEL and XPDL process definitions of different dimensions using BPMN model presented in particular dimension – this means that the modeler will create BPMN business process model in one dimension and save the XPDL or BPEL definitions in other dimensions for further import to another tool or execution.

In the situations where business process model contains several executable pools (B2B process), BPEL is created for each pool separately. The future work will address the issue of generating BPEL code from collaboration BPMN business process model, which contains message flows that cross the boundaries of the pools and how the transformation of collaboration process to another dimension affects BPEL process definitions of each pool involved in the process and the interactions between these pools.

BPMN, BPEL and XPDL are important standards in the BPM industry, often referred as „BPMN-XPDL-BPEL value chain" but the translating between these standards is a challenging task, as there exist only non- normative mapping guidelines, which may differ from vendor to vendor. BPMN, BPEL and XPDL are complimentary standards, which using without each other represent no value to BPM.

REFERENCES

[1] C. Ouyang, W. M. P. van der Aalst, M. Dumas, and A. H. M. ter Hofstede, "From business process models to process-oriented software systems: the BPMN to BPEL way," 2006.
[2] "Business process modeling notation (BPMN) specification," Object Management Group. [Online]. Available: http://www.bpmn.org.
[3] "XML process definition language (XPDL) specification," Workflow Management Coalition. [Online]. Available: http://www.wfmc.org.
[4] B. J. Matjaz and P. Kapil, *Business Process Driven SOA using BPMN and BPEL*. Packt Publishing, 2008.
[5] "Web services business process execution language version 2.0 specification," OASIS Open. [Online]. Available: http://www.oasis-open.org/committees/download.php/10347.
[6] A. W. Stephen, *XPDL and BPMN*, SeeBeyond, United States, 2003.
[7] L. Penicina, "The approach of transformation between business process dimensions in BPMN modeling tool," in *The 15th International Conference on Information and Software Technologies (IT 2009)*, 2009, pp. 72-81.
[8] M. Kirikova, L. Businska, and L. Penicina, "Multidimensional business process modeling" in *The 7th International Conference on Perspective in Business Informatics Research (BIR 2008)*, 2008, pp. 196-210.
[9] "Oracle BPEL Process Manager," Oracle Corporation. [Online]. Available: www.oracle.com/technology/bpel.
[10] "Microsoft BizTalk Server 2004," Microsoft Corporation. [Online]. Available: www.microsoft.com/BizTalk.
[11] "Intalio|BPMS 5.2," Intalio. [Online]. Available: http://community.intalio.com.
[12] Object Management Group. [Online]. Available: www.omg.org.
[13] "Advancing Open Standards for the Information Society," OASIS. [Online]. Available: http://www.oasis-open.org.
[14] L. Penicina, "The mapping of multidimensional BPMN models to BPEL," in *Cases and Projects in Business Informatics – International Business Informatics Challenge 2009, Third International Business Informatics Challenge and Conference 2009*, Ireland, Dublin, 25 September, 2009.

Ludmila Penicina was born in June 1, 1986 in Rezekne, Latvia. In year 2008 she graduated Riga Technical University, Riga, Latvia with bachelor's degree of engineering science in computer control and computer science.

Author currently is 2nd year master degree student in Faculty of Computer Science and Information Technology, Institute of Applied Computer Systems, Riga Technical University, Riga, Latvia and 2nd year MBA student in Riga International School of Economics and Business Administration, Riga, Latvia. She is currently also working as a Scientific Assistant in Faculty of Computer Science and Information Technology, Institute of Applied Computer Systems and as a System Analyst in Information Technology department, Riga Technical University, Riga, Latvia. During June 2007 – May 2008 she used to work as a Tester Specialist in Lattelecom Technology, Riga, Latvia.

Author has received the International Business Informatics Challenge 2009 (IBIC'09) Best Paper Award for the contribution on "The Mapping of Multidimensional BPMN models to BPEL."

Ludmila Peņicina. Daudzdimensiju BPMN modeļu kartēšana procesu aprakstu standartos
Biznesa procesu vadības (BPV) aprindās Biznesa procesu modelēšanas notācija (BPMN) ir apstiprināta kā de facto biznesa procesu modelēšanas standarts. Šajā rakstā BPMN ir izvēlēta kā notācija daudzdimensiju biznesa procesu modelēšanai, jo BPMN metamodelis ļauj papildināt grafiskos notācijas elementus ar specifiskiem atribūtiem, kas ļauj aprakstīt svarīgākos biznesa procesa aspektus, ko var uzskatīt par modelēšanas dimensijām. Taču, BPMN ir tikai grafiskā notācija un pilnībā neatbalsta pilno biznesa procesa dzīves ciklu – sākot ar dokumentēšanu un modelēšanu un beidzot ar procesu izpildīšanu un monitoringu. Lai sekmētu BPV popularitāti un ātrāku akceptēšanu organizāciju informācijas sistēmu izstrādē, industrijā parādījās vēl divi BPMN papildinošie standarti – BPEL un XPDL. Šie standarti ir bāzēti uz XML, taču tiem ir pilnīgi dažādi uzdevumi – BPEL ir standarts procesu palaišanai un izpildīšanai, savukārt, XPDL ir vienots standarts BPMN modeļu uzglabāšanai un importam/eksportam BPMN rīkos. Ir veikti dažādi pētījumi, kuros ir aprakstīti jautājumi un problēmas, kas saistās ar BPMN kartēšanu uz XPDL un BPEL standartiem. Šā pētījuma mērķis ir noskaidrot, kā mainās BPEL un XPDL procesu definīciju struktūra, pēc BPMN modeļa transformācijas citās dimensijās.

Лудмила Пеницина. Преобразование многомерных BPMN моделей в стандарты описания процессов
Общество управления бизнесс процессов приняло BPMN как де-факто стандарт для моделирования бизнесс процессов. В этом исследовании BPMN выбран как основной стандатр для представления многомерных деловых процессов, так как он состоит из набора графических элементов, которые могут быть дополнены специальными атрибутами, позволяя описать каждый важный аспект бизнесс процесса – который может быть расценен, как измерение процесса. Однако, BPMN является только графическим изображением и не поддерживает управление бизнесс процессами – полную поддержку жизненного цикла бизнесс процесса: от документации и моделирования к выполнению и контролю. Чтобы ускорить принятие управления бизнесс процессами, появились два стандарта дополняющие BPMN – Язык Выполнения Делового Процесса (BPEL) и Язык Определения Процесса XML (XPDL). Эти два стандарта используются для различных целей – BPEL язык выполнения процесса и XPDL - язык обмена процесса. Проблема преобразование моделей BPMN на BPEL и XPDL описываеться в различных источниках. Данное иследование изучает как меняется структура BPEL и XPDL для BPMN моделей в различных измерениях.