Survey of UML Based XPDL Document for Business Process Modeling

Sang-Young Lee

Department of Health Administration, Namseoul University, Cheonan, South Korea
sylee@nsu.ac.kr

Abstract

Currently there are a variety of different tools that may be used to analyze, model, describe and document a business process. However, it is difficult to exchange the information of a business process created in different tools because of the distinct information formats used in different tools. The XML Process Definition Language(XPDL) of the Workflow Management Coalition(WfMC) forms a common interchange standard that enables products to continue to support arbitrary internal representations of process definitions with an import or export function to map to or from the standard at the product boundary. Generally a business process model can be represented by the UML activity diagram, but there is a difficult task to directly generate an XPDL document from a business process model represented by the standard activity diagram. In the paper we will propose an approach to generate an XPDL document from a business process model represented by the extended UML activity diagram and provide an implementation for the approach.

Keywords: *XPDL, Business Process Definition, UML*

1. Introduction

The XPDL specification uses XML as the mechanism for process definition interchange. A Process Definition is defined as: The representation of a business process in a form that supports automated manipulation, such as modeling, or enactment by a workflow management system. The process definition consists of a network of activities and their relationships, criteria to indicate the start and termination of the process, and information about the individual activities, such as participants, associated IT applications and data, etc [1]. The process definition is expressed in a consistent form, which is derived from the common set of objects, relationships and attributes expressing its underlying concepts, so that a variety of different tools can use the process definitions as interchange media to exchange the information from each other. The principles of process definition interchange are illustrated in Figure 1.

The UML activity diagram also can create in forms of sets of activities and transitions between them [2, 3, 4]. So it is possible to map the business process model, which is represented by UML activity diagram, to the process definition organized in XPDL. However, it is difficult to directly map the business process model represented by the standard UML activity diagram to the process definition organized in XPDL because some elements in the standard UML activity diagram cannot be directly associated to entities defined in the process definition. Hence the paper will propose a method of generating an XPDL document from a business process model represented by an extended activity diagram. For the purpose of this paper, the terms process definition, business process model, and workflow model are all considered to represent the same concept, and therefore, they are used interchangeably.

The paper consists of five sections. The following section discusses issues for relate works. The third section describes the XPDL document structure and entity definitions associated to modeling elements of UML activity diagram. The method of a mapping from the business process model represented by extended activity diagram to the XPDL document will be depicted in the fourth section. Section 5 presents our conclusion and future work. The references will be listed in the end.

2. Issues for Related Works

Currently a tool for business process modeling has been implemented, which could create extended activity diagram for modeling business processes or workflows. In order to make the generated business process model available for other modeling tools or workflow systems, there must be a common interchange standard existing for information exchange among the diverse tools.

Fortunately XPDL uses XML as the mechanism for process definition interchange. XPDL specification provides the workflow process definition interface that defines a common interchange format, which supports the transfer of workflow process definitions between separate products. The interface also defines a formal separation between the development and run-time environments, enabling a process definition, generated by one modelling tool, to be used as input to a number of different workflow run-time products[5]. A business process model can be expressed conveniently by using the UML activity diagram. Now the key problem is how to generate a XPDL document from a business process model represented by UML activity diagram. A solution will be proposed in the fourth section.

The following section will briefly describe the XPDL document structure and the definitions of relevant XPDL entities related to modeling elements of UML activity diagram [6].

3. XPDL Document Structure and Associated Entities

The content of a XPDL document mainly describes Process Definition(s). The Figure 2 shows the top-level entities contained within a Process Definition [7].

- Workflow Process Definition: The Process Definition entity provides contextual information that applies to other entities within the process. It is a container for the process itself and provides information that associate with administration (creation data, author, etc.) or to be used during process execution (initiation parameters to be used, execution priority, time limits to be checked, person to be notified, simulation information, etc.).

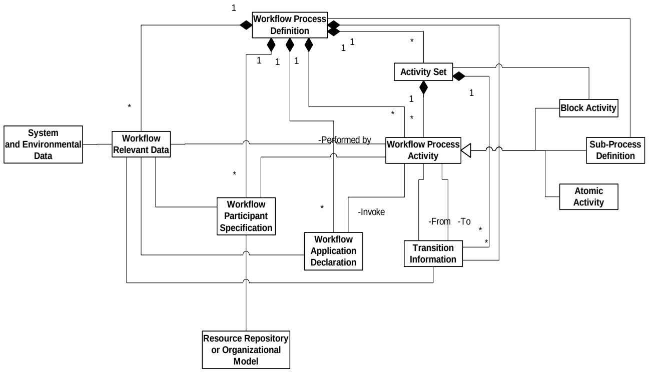

Workflow Process Definition
1
1
1
1
1
1
*
Activity Set
1
1
*
*
*
Block Activity
System and Environmental Data
Workflow Relevant Data
-Performed by
Workflow Process Activity
Sub-Process Definition
*
Atomic Activity
Workflow Participant Specification
*
-Invoke
-From
-To
*
*
Workflow Application Declaration
Transition Information
Resource Repository or Organizational Model

Figure 1. Meta-Model Top-Level Entities

- Workflow Process Activity: A process definition consists of one or more activities, each comprising a logical, self-contained unit of work within the process. An activity represents work, which will be processed by a combination of resource specified by participant assignment and/or computer applications specified by application assignment.

- Transition Information: Activities are related to one another via flow control conditions. Each individual transition has three elementary properties, the from-activity, the to-activity and the condition under which the transition is made. Transition from one activity to another may be conditional (involving expressions which are evaluated to permit or inhibit the transition) or unconditional. The transitions within a process may result in the sequential or parallel operation of individual activities within the process.

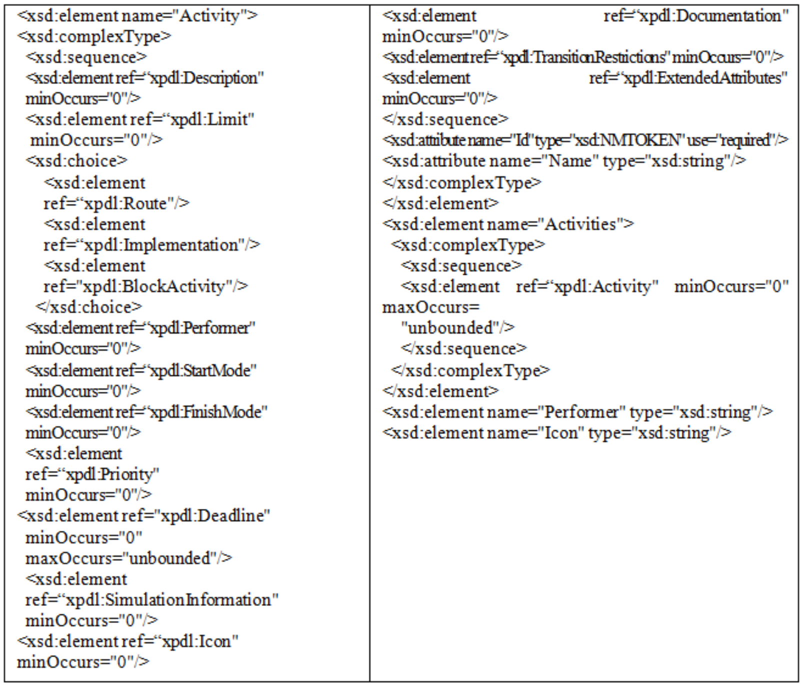
- Workflow Participant Declaration: This provides descriptions of resources that can act as the performer of the various activities in the process definition.

- Resource Repository: The resource repository for the fact that participants can be humans, programs, or machines.

- Workflow Application Declaration: This provides descriptions of the IT applications or interfaces which may be invoked by the workflow service to support, or wholly automate, the processing associated with each activity, and identified within the activity by an application assignment attribute(s).

- Workflow Relevant Data: This defines the data that is created and used within each process instance during process execution.

- System and Environment Data: This is data which is maintained by the workflow management system or the local system environment, but which may be accessed by workflow activities or used by the workflow management system in the evaluation of conditional expressions in the same way as workflow relevant data.



```
<xsd:element name="Activity">
<xsd:complexType>
 <xsd:sequence>
 <xsd:element ref="xpdl:Description"
 minOccurs="0"/>
 <xsd:element ref="xpdl:Limit"
  minOccurs="0"/>
 <xsd:choice>
    <xsd:element
    ref="xpdl:Route"/>
    <xsd:element
    ref="xpdl:Implementation"/>
    <xsd:element
    ref="xpdl:BlockActivity"/>
   </xsd:choice>
 <xsd:element ref="xpdl:Performer"
 minOccurs="0"/>
 <xsd:element ref="xpdl:StartMode"
 minOccurs="0"/>
 <xsd:element ref="xpdl:FinishMode"
 minOccurs="0"/>
 <xsd:element
 ref="xpdl:Priority"
 minOccurs="0"/>
<xsd:element ref="xpdl:Deadline"
 minOccurs="0"
 maxOccurs="unbounded"/>
 <xsd:element
 ref="xpdl:SimulationInformation"
 minOccurs="0"/>
<xsd:element ref="xpdl:Icon"
minOccurs="0"/>
```

```
<xsd:element                    ref="xpdl:Documentation"
minOccurs="0"/>
<xsd:element ref="xpdl:TransitionRestrictions" minOccurs="0"/>
<xsd:element                    ref="xpdl:ExtendedAttributes"
minOccurs="0"/>
</xsd:sequence>
<xsd:attribute name="Id" type="xsd:NMTOKEN" use="required"/>
<xsd:attribute name="Name" type="xsd:string"/>
</xsd:complexType>
</xsd:element>
<xsd:element name="Activities">
 <xsd:complexType>
   <xsd:sequence>
   <xsd:element   ref="xpdl:Activity"   minOccurs="0"
maxOccurs=
   "unbounded"/>
   </xsd:sequence>
 </xsd:complexType>
</xsd:element>
<xsd:element name="Performer" type="xsd:string"/>
<xsd:element name="Icon" type="xsd:string"/>
```

Figure 2. XPDL Schema of “Process Activity” Element

- Data Types and Expressions: The meta-model (and associated XPDL) assumes a number of standard data types (string, reference, integer, float, date/time, etc.); such data types are relevant to workflow relevant data, system or environment data or participant data.

According to the descriptions of the top-level entities in process definition, a business process model represented by UML activity diagram can be mapped into the elements contained both in "Workflow Process Activity"entity and in "Transition Information" entity. The following figure shows the XPDL schema structure of the process activity element, and the table 1 shows the descriptions of attributes of process activity element.

Table 1. Process Activity – Attributes

Attributes	Descriptions
BlockActivity	An Activity that executes an ActivitySet.
Deadline	Specification of a deadline and action to be taken if it is reached.
Description	Textual description of the activity.
Documentation	The address (e.g. path- and filename) for a help file or a description file of the activity.
Extended Attributes	Optional extensions to meet individual implementation needs
Finish Mode	Describes how the system operates at the end of the Activity.
Icon	Address (path- and filename) for an icon to represent the activity.
Id	Used to identify the workflow process activity.
Implementation	A "regular" Activity. Mandatory if not a Route. Alternative implementations are "no", or "subflow".
Limit	Expected duration for time management purposes (e.g. starting an escalation procedure etc.) in units of DurationUnit. It is counted from the starting date/time of the Process. The consequences of reaching the limit value are not defined in this document (i.e. vendor specific).
Name	Text Used to identify the workflow process activity.
Performer	Link to entity workflow participant. May be an expression. Default: Any Participant.
Priority	A value that describes the initial priority of this activity when it starts execution. If this attribute is not defined but a priority is defined in the Process definition then that is used. By default it is assumed that the priority levels are the natural numbers starting with zero, and that the higher the value the higher the priority (i.e.: 0, 1,…, n).
Route	A "dummy" Activity
Simulation Information	Estimations for simulation of an Activity. No default.
Start Mode	Describes how the execution of an Activity is triggered.
Transition Restrictions	Provides further restrictions and context-related semantics description of Transitions

The following figure shows the XPDL schema structure of transition shows the descriptions of attributes of Transition element.

```
<xsd:element name="Transition">
<xsd:complexType>
<xsd:sequence>
<xsd:element ref=“xpdl:Condition" minOccurs="0"/>
<xsd:element ref=“xpdl:Description" minOccurs="0"/>
<xsd:element ref=“xpdl:ExtendedAttributes" minOccurs="0"/>
</xsd:sequence>
<xsd:attribute name="Id" type="xsd:NMTOKEN" use="required"/>
<xsd:attributename="From" type="xsd:NMTOKEN" use="required"/>
<xsd:attribute name="To" type="xsd:NMTOKEN" use="required"/>
<xsd:attribute name="Name" type="xsd:string"/>
</xsd:complexType>
</xsd:element>
<xsd:element name="Transitions">
<xsd:complexType>
<xsd:sequence>
<xsd:element ref=“xpdl:Transition" minOccurs="0" maxOccurs=
"unbounded"/>
</xsd:sequence>
</xsd:complexType>
</xsd:element>
```

Figure 3. XPDL Schema of “Transition” Element

Table 2. Transition Information – Attributes

Attributes	Descriptions
Condition	A Transition condition expression based on workflow relevant data. (E.g. 'Contract' = 'SMALL' OR 'Contract' <$20,000). Default: TRUE
Description	Short textual description of the Transition.
Extended Attributes	Optional extensions to meet individual implementation needs
From	Determines the FROM source of a Transition. (Activity Identifier)
Id	Used to identify the Transition.
Name	Text used to identify the Transition.
To	Determines the TO target of a Transition (Activity Identifier)

4. Mapping from Business Process Model to XPDL Document

The task of the mapping from business process model represented by the UML activity diagram to the corresponding XPDL document actually is to generate the corresponding information in the format of XPDL from each element in source business process model and put the information into the appropriate positions in XPDL document structure. Considering that one “activity” element in UML activity diagram can not be definitely associated to one entity in XPDL, we just refine activity into three different types of activities which can be directly related to the ‘regular’ three different implementation types of activities respectively.

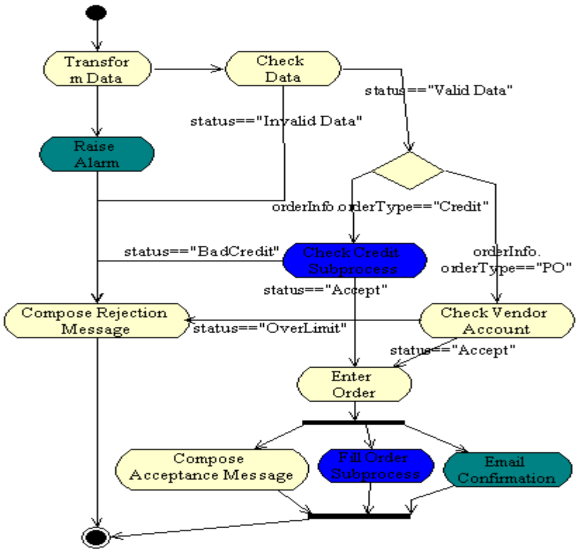


Figure 4 : EOrder Process

The above activity diagram presents a process called ‘EOrder’ which is used to present an electrical order process. The process takes a formatted string as an input and returns a string that indicates whether the order was confirmed or rejected. In the subsequent part we will use the business process model as an example to illustrate the mapping method.

4.1 The Mapping of “Start State” Element and “End State” Element

The “start state” and “end state” element can be mapped to the “Route activity” entity defined in XPDL.The Route Activity is a “dummy” Activity that permits the expression of “cascading” Transition conditions (e.g. of the type "IF condition-1 THEN TO Activity-1 ELSE IF condition-2 THEN TO Activity-2 ELSE Activity-3 ENDIF"). A route activity has neither a performer nor an application and its execution has no effect on workflow relevant data or application data.

<Activity Id="5"> <Description> This is a start!</Description> <Route/> <ExtendedAttributes> <ExtendedAttribute Name="Coordinates"> <xyz:Coordinates xpos="35" ypos="389"/> </ExtendedAttribute> </ExtendedAttributes> </Activity> 6-1 The XPDL of The “Start state” Element <Activity Id="6"> <Description> This is an end! </Description>	<Route/> <TransitionRestrictions> <TransitionRestriction> <Join Type="XOR"/> </TransitionRestriction> </TransitionRestrictions> <ExtendedAttributes> <ExtendedAttribute Name="Coordinates"> <xyz:Coordinates xpos="755" pos="315"/> </ExtendedAttribute> </ExtendedAttributes> </Activity> 6-22 The XPDL of The “End state” Element

Figure 5. XPDL Representations of “Start State” and “End State” in the “EOrder” Process

4.2 The Mapping of Activity Element

Activity elements with three different colours of green, yellow and blue can be mapped to “regular” activity entities with three different implementation types which are defined in XPDL: “No Implementation”, “Tool”, and “Subflow” respectively. “No Implementation” type means that the implementation of this activity is not supported by workflow using automatically invoked applications or procedures. “Tool” type means that the activity is implemented by (one or more) tools. A tool may be an application program. “Subflow” type means that the activity is refined as a subflow. A subflow may be executed synchronously or asynchronously.

4.3 The Mapping of Synchronization Element

A synchronization element can be mapped to a route activity with “Transition Restriction” attribute with kind of “join” or “spit”, which is defined in XPDL. A join corresponding to a “join” kind synchronization element describes the semantics of an activity with multiple incoming transitions. A split corresponding to a “fork” kind synchronization describes the semantics where multiple outgoing transitions for an activity exist. Both join and spit have the types of “AND” and “XOR”. Figure 8 shows XPDL representations of a “fork” kind synchronization element and a “join” kind synchronization element in the “EOrder” process.

4.4 The Mapping of Decision Element

A decision element can be mapped to a route activity entity with transition restriction of spit kind with XOR type, which is defined in XPDL. Figure 9 shows the XPDL representation of “Check Order Type” decision element in the “EOrder” process. In the XPDL representation “TransitionRef” attribute indicates that “Check Order Type” has two outgoing transitions whose Ids are 24 and 25 respectively in the “EOrder” process model.

```
<Activity Id="12" Name=
"Check Order Type">
<Route/>
<TransitionRestrictions>
<TransitionRestriction>
<Split Type="XOR">

<TransitionRefs>
<TransitionRef Id="24"/>
<TransitionRef Id="25"/>
</TransitionRefs>
</Split>
</TransitionRestriction>
</TransitionRestrictions>
<ExtendedAttributes>
<ExtendedAttribute Name="Coordinates">
<xyz:Coordinates xpos="293" pos="460"/>
</ExtendedAttribute>
</ExtendedAttributes>
</Activity>
```

Figure 6. XPDL Representation of “Check Order Type” Decision Element in the “EOrder” Process

4.5 The Mapping of Transition Element

A transition element can be mapped to a transition entity defined in XPDL. The transition entities describe possible transitions between activities and the conditions that enable or disable them (the transitions) during workflow execution. The following figure shows XPDL representations of all transition elements in the “EOrder” process.

```
<Transitions>
  <Transition Id="1" From="9" To="8"/>
  <Transition Id="2" From="9" To="11"/>
  <Transition Id="16" From="11" To="33"/>
  <Transition Id="17" From="8" To="33">
  <Condition Type="OTHERWISE"/>
</Transition>
<Transition Id="18" From="33" To="6"/>
<Transition Id="20" From="5" To="17"/>
<Transition Id="21" From="17" To="1"/>
<Transition Id="22" From="1" To="12">
<Condition>status
== "Valid
Data"</Condition>
</Transition>
<Transition Id="23" From="1" To="39">
<Condition>status == "Invalid Data"
</Condition>
</Transition>
<Transition Id="24" From="12" To="10">
<Condition>orderType == Credit"</Condition>
</Transition>
<Transition
Id="25" From="12" To="41">
<Condition>orderType == "PO"</Condition>
</Transition>
```

```
<Transition Id="26" From="10" To="32">
<Condition>status == "Accept"</Condition>
</Transition>
<Transition Id="27" From="41" To="32">
<Condition>status == "Accept"</Condition>
</Transition>
<Transition Id="28" From="32" To="9"/>
<Transition Id="29" From="39" To="6"/>
<Transition Id="30" From="41" To="39">
<Condition>status == "OverLimit"</Condition>
</Transition>
<Transition Id="31" From="10" To="39">
<Condition>status == "BadCredit"</Condition>
</Transition>
<Transition Id="38" From="9" To="56"/>
<Transition Id="39" From="56" To="33"/>
<Transition Id="40" From="17" To="58">
<Condition Type="EXCEPTION"/>
</Transition>
<Transition Id="42" From="58" To="39"/>
</Transitions>
```

Figure 7. XPDL Representations of All Transition Elements in the "EOrder" Process

5. Conclusions

We In the paper we proposed the approach of the mapping from the business process model represented by extended activity diagram to XPDL document and provide the implementation for that approach.

The approach proposed in this paper provides the important guidance for the mapping from other similar business process models, which are used in different software products, to XPDL document. The approach can successfully generate XPDL document from business process model presented by UML activity diagram. However, not all the modelling elements of UML activity diagram can be mapped into corresponding XPDL entities. We still haven't found an approach to map the "Swimlane" element of UML activity diagram into the XPDL entity because of the lack of associated XPDL entities. In the future, we will do much more research and try to seek an approach to solve the problem. Now the implementation still can not generate the complete XPDL document from business process model presented by UML activity diagram because the XPDL document contains some data information that can not be obtained directly from the UML activity diagram. Therefore, In the future it is necessary to improve the implementation for the complete XPDL document generation.

References

[1] A. Alexander, "A Co-Operative Task Modeling Approach to Business Process Understanding", Workshop on Object-Oriented Business Process Modeling, ECCOOP, **(1998)**.

[2] S. Aissi, P. Malu and K. Srinivasan, "E-business Process Modeling: The Next Big Step", Computer, vol. 35, Issue. 5, **(2010)**, pp. 55-62.

[3] H. –E. Eriksson and M. Penker, "Business Modeling with UML: Business Patterns and Business Objects", John Wiley & Sons Inc., **(2008)**.

[4] J. Heumann, "Introduction to Business Modeling Using the Unified Modeling Language(UML)", Rational Software, **(2001)**.

[5] Object Management Group(OMG), "OMG Unified Modeling Language Specification", http://www.rational.com, **(2010)**.

[6] O. Wiegert, "Business Process Modeling & Workflow Definition with UML", SAP AG, **(1998)**.

[7] Workflow Management Coalition(WfMC), "Workflow Process Definition Interface - XML Process Definition Language(XPDL) Specification", http://www.wfmc.org, **(2009)**.

Authors

Sang-Young Lee

Professor, Dept. of Health Administration, Namseoul University, South Korea.