XPDL 2.0: Integrating Process Interchange and BPMN

Robert M. Shapiro, Global 360, USA

HISTORY

The basic concepts that underlie XPDL\(^1\) were formulated by individuals working together in the WfMC\(^2\) who were from companies developing workflow and business process management (BPM) tools. These concepts were embodied in a meta-model and glossary which then guided the specification of interfaces for various aspects of the overall problem. The interchange of process definitions between different tools and also different vendors was regarded as an essential piece of this whole and the first version of a standard interchange language was the Workflow Process Definition Language (WPDL), published by the WfMC in November 1998.

The growing popularity of XML and its use for defining document formats for the Internet, combined with some years of accumulated experience using WPDL in workflow and BPM tools, led to the creation of XPDL 1.0, which was officially released in October 2002. XPDL retained the semantics of WPDL but defined a new syntax using an XML schema. Neither WPDL nor XPDL 1.0 proposed a specific graphical representation, although the underlying meta-model for a process was based on a directed graph structure consisting of activities as nodes and transitions as the edges or pathways between them.

The Business Process Management Notation (BPMN) was developed by individuals working together in the Business Process Management Initiative (BPMI.org\(^3\)) to take the techniques employed in flowcharting tools, unify and extend the graphics to express the semantics required in workflow and EAI processes. BPMN 1.0 was released in May 2004.

In addition to the graphical notation, BPMN incorporated a number of specific mechanisms for process modeling that had not yet been included in XPDL; among these in particular events and message passing between processes. XPDL 2.0 incorporates these mechanisms as well as the graphics and offers an extended meta-model that unifies XPDL and BPMN. It was officially approved by the WfMC membership in October 2005.

BASICS

A Business Process Model consists of a collection of processes together with the applications and resources required to perform all the steps contained in the processes. We start with a discussion of the elements in a single process. To keep things simple we focus first on a minimal subset of elements.

- Activities (the steps in the process)
- Transitions (or sequence flow)

---

\(^1\) XML Process Definition Language  
\(^2\) Workflow Management Coalition  
\(^3\) BPMI.org merged with the Object Management Group (OMG) in June 2005.
For a complete description of the graphics, syntax and semantics refer to the websites of BPMI.org\textsuperscript{4} and WfMC.org.\textsuperscript{5}

In the following picture the four round rectangles are activities and the four arrows are transitions which define possible paths between activities. The leftmost activity, Receive Payment, is the first activity to be performed in the process since it has no predecessor. Send Receipt is the last activity. Process Large Payment and Process Small Payment are alternatives: a payment is routed to one or the other according to the amount of the payment.

The Boolean expressions on the transitions determine which transition occurs, as determined by a piece of data, \textit{amount}, associated with the payment. This notation is commonly employed in Petri Nets, a mathematical formalism that influenced the development of XPDL.

Flow charts often use specific graphical elements to indicate branching logic. To support this XPDL offers routing activities (gateways). The diagram could then be drawn as:

Here the diamond shaped nodes are routing activities and the X indicates exclusive-or logic. The XOR node on the left is a branch. Only one exit transition has a condition expression; the other path is marked as the default gate, to be taken only if the conditions on all other exit gates evaluate to \textit{false}.

\textsuperscript{4} BPMI.org; Business Process Modeling Notation (BPMN), version 1.0 – May 3, 2004
\textsuperscript{5} WfMC.org: XPDL 2.0 WFMC-TC-1025 Version 1.0 October 3, 2005
There are a number of gateway types. For the most commonly used are:

- **Exclusive (XOR)**
- **Parallel (AND)**

The nodes and transitions can form arbitrarily complex graphs with:
- Sequential Activities
- Parallel Activities
- Loops / Cycles
- Conditional Paths

In the above flow we have introduced a third type of activity node, an Event. An event is something that “happens” during the course of a business process. These events affect the flow of the process and usually have a cause (trigger) or an impact (result). There are three types of Events, based on when they affect the flow: Start, Intermediate, and End. There are numerous trigger and result types. Some common combinations are:

- **Start Event** triggered by a message.
- **Intermediate Event** triggered by a Timer.
- **End Event** with Error Result

**Activities**

We have said nothing about the details of a basic activity. It is a step in the process, but what happens at that step? Typically, to perform a step requires one or more resources: e.g. a person with a particular skill set or a system resource. A task or application may need to be executed to perform the step.
So basic activities have attributes which provide information about who can perform the activity, what application or web services should be invoked, what properties of the object being worked on are used and/or altered in this step, and so forth.

The participants (resources) and applications may be defined within a single process or for the entire collection of processes in the Business Process Model. The properties of work objects are likewise definable within a single process or for the entire model.

Activities have other attributes which further define their specific role or how they are implemented. Here we list a few:

- **StartActivity** Indicates that process starts here.
- **IsATransaction** Transaction-based semantics.

**Tasks and Applications**

There are seven standard Tasks that can be specified for a basic Activity and are used primarily for invoking Web Services and using WSDL\(^6\) messaging. An eighth task type is used for invoking Applications whose signatures have been defined in the Business Process Model.

Applications that can be invoked by a process are defined at the Process or Package (See Package meta-model) level. There are multiple types of applications:

- Traditional applications
- Components
- Web Services
- Business Rules
- Form
- Script

**Compound Activities**

A compound activity refers to another process (independent or embedded). The graphics for a compound activity include a special marker to designate this.

Compound activities allow re-use of process definitions. If the reference is to an independent sub-process, a list of actual parameters can be passed to the sub-process at the time of invocation. An embedded sub-process shares the same data space so no parameters are passed. (As a technical detail not shown in the graphical representation, an embedded process is an XPDL activity set and the compound activity is referred to as a Block Activity). An independent sub-process may be invoked asynchronously, in which case the exit transitions from the compound activity immediately determine the routing of the work object to the successor of the compound activity. Synchro-
nous invocation requires that the sub-process complete before the compound activity can continue. Embedded sub-processes are always invoked synchronously.

**PROCESS META MODEL**

The meta-model depicts the relationships between all the elements in a Process. The shaded elements were not present in XPDL 1.0 and are now included in XPDL 2.0 to support BPMN constructs. Gateway and Event have already been described. Pool and Lane are elements that support the use of graphical Swimlanes which we discuss later in this chapter.
This meta-model describes the relationship between elements on the Package or Business Process Diagram level. The shaded elements were not present in XPDL 1.0 and are now included in XPDL 2.0 to support BPMN constructs.
**SWIMLANES**

In the following diagram we depict a single pool containing the process *Loan Application*. There are no lanes in the pool. The process can be either an independent sub-process or an embedded sub-process.

In the next diagram we depict a single pool containing the process *Loan Application*. This pool has two lanes. An activity in a lane requires as performer the resource designated by the name of the lane.

Notice that the transitions (sequence flow) may go across lanes in a pool. Transitions may not go across pools.

**MESSAGE FLOW**

Message Flow is normally implemented by Web Services and Message Queues.

In the example we illustrate how message flow may go between activities in different pools. This allows us to represent graphically aspects of the choreography between processes. It should be noted that message flow cannot occur between activities in the same pool. In other words, sequence flow is used to connect activities in the same pool whereas message flow is used to represent communication between activities in different pools.

In these examples pools have been drawn with a horizontal orientation and a width that extends across the entire page. However, the specification supports vertical pools as well, and also allows the width/height to be limited. This supports the use of pools in the specification of abstract processes and their choreography.
ARTIFACTS

Artifacts provide documentation facilities with graphical representation. Associations are used to connect the artifacts to flow objects such as activities. The association is like a sequence or message flow, but allows optional arrows at either or both ends of the pathway.

Data Object:

Text Annotation:

Group: Allows diverse objects to be labeled with the same name for reporting purposes.
**PROCESS DEFINITION AND REPORTING**

An XPDL 2.0 document contains the process definitions for a collection (Package) of processes. This XML document is used not only by modeling tools, simulation tools and execution (enactment) engines; it also provides the basic information for BAM\(^7\) reporting tools and in particular provides the dimensions and members for OLAP\(^8\) cube reporting technologies.

Here we depict a Business Process Management System which uses the Admin facility to send the XPDL process definitions to the analysis engine and transmits a stream of log events which capture the details of execution. The Analysis Engine structures the database and OLAP cubes based on the process definitions, participant and queue information. The events are processed by the Analysis Engine to update the fact and dimension tables in the database and cube processing completes the preparation for interactive slice and dice viewing of the data, using EXCEL and/or other proprietary Process and Business Intelligence tools. Below is an EXCEL chart based on historical data.

---

\(^7\) Business Activity Monitoring

\(^8\) OnLine Analytical Processing
An alternate approach to data presentation shows selected data in the visual context of the process definition. This can be done for historical presentations as well as animations of the executing system or a simulation run.

**Simulation**

Simulation engines based on the WfMC meta-model and the XPDL file format have been available for a number of years. These engines are driven by XML scenario files. The XPDL package is supplemented by a number of additional schemata that provide information required for simulation, including details about the performance of the activities (e.g., duration information), schedules for the arrival of work, resource characteristics (skill sets, work

---

**Work Type** | **Started** | **Completed** | **Cycle Time** | **Cost**
---|---|---|---|---
Leads OR | 130 | 20 | 22.5 | $120.00
Leads SE | 100 | 40 | 40 | $180.00
Leads NC | 87 | 5 | 10 | $200.00
Leads SC | 40 | 10 | 25 | $100.00
schedules, cost etc.) and a variety of simulation options. Some of this data can be acquired automatically from historical information collected in the OLAP data base.

These simulators are able to generate log event streams identical to those produced by the BPM execution engine. Hence the same Analysis engine can be utilized to provide charts and reports that evaluate changes being tested using simulation.

The simulation technology needs to be extended to include new constructs incorporated in XPDL 2.0; in particular the BPMN events and message flow.

**PROCESS INTERCHANGE**

Common meta-model allows tools to exchange models.

Type of tools:
- Simulation tools
- Monitoring tools
- Execution tools
- Modeling tools
- Repository tools

The following diagram illustrates the use of process interchange in a BPM suite.
CONCLUSION

XPDL 2.0 provides a standard graphical approach to Business Process Definition based on BPMN graphics. XPDL 2.0 provides a standard file format for persisting BPMN diagrams and interchanging Process definitions. The file format is based on the WfMC meta-model which establishes a framework for defining, importing and exporting process definitions for numerous products including execution engines, simulators, BPA modeling tools, Business Activity Monitoring and reporting tools.

The schema defining the format is extensible and provides vendor and user extension capabilities as well as a natural path for future versions of the standard. Mappings to specific execution languages (e.g. BPEL) and other XML-based specifications (e.g. ebXML) are possible.
Analysis Engine, 191
BPMI.org, 184
BPMN, 183, 184, 187, 188, 192, 193
Business Process Diagram, 188
Business Process Management Notation (BPMN), 183
Business Process Model, 183
embedded sub-process, 186

OLAP, 191, 192
Petri Nets, 184
simulation technology, 192
Swimlanes, 187
Web Services, 186
WPDL, 183
XPDL, 183, 184, 186, 187, 188, 191, 192, 193