A Format for a Process Design Ecosystem

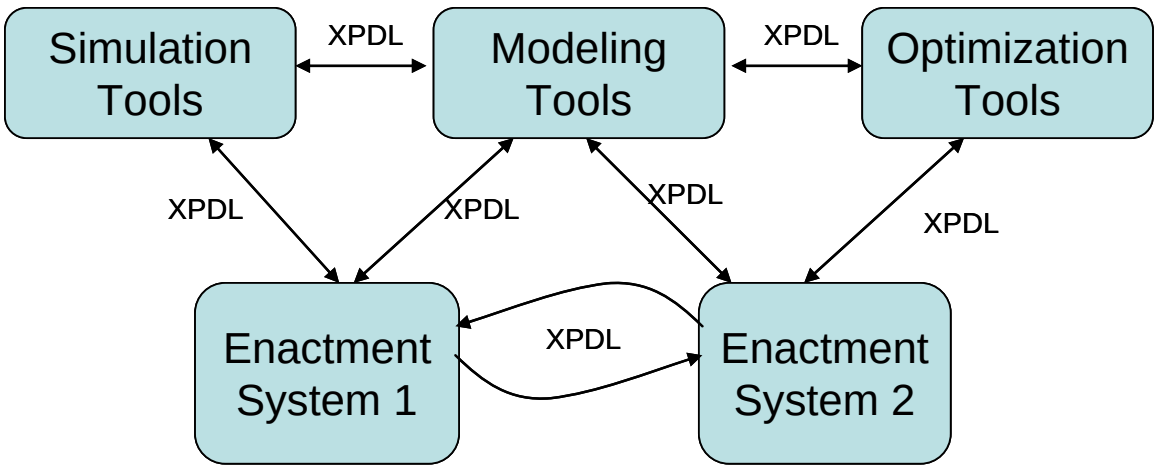
Keith Swenson

The decade of the 90s was a time to get people thinking about business in terms of a "process." We were more successful than many thought possible. No business analysts today would approach optimization of their business or organization without first considering the main processes involved.

Along with this change in thinking come a wide variety of software tools to help us reach business process optimization. Among the modeling tools are high-level corporate goal modeling tools, macro-level production flow tools, and personal and workgroup-oriented coordination tools, as well as a few systems that cover the complete range of modeling options. Beyond modeling tools, there are also simulation tools, optimization tools, analytical tools, enactment tools, monitoring tools, documentation tools, and conversion tools. This is expected in a field that is rapidly growing.

Process Design Interchange

Typically, as new technologies are adopted, different specialists in an organization tend to adopt different best-of-breed tools from different vendors, but as the process technology market matures, customers have begun to demand that these tools interoperate. They are no longer satisfied starting with a blank sheet and manually entering the data. For example, some specialists may want to run simulations, but, instead of drawing the process from scratch, they want to import the process from their process design tool. Or they may be ready to implement a process application, and they want to start by importing the process from the high-level modeling tool. Today, there are many vendor-specific interchange formats for companies standardizing on a single vendor, but for companies that want to be able to use best-of-breed solutions from different vendors, there is a clear need for an open, non-proprietary Process Design Ecosystem that allows users to *exchange process designs* from one vendor's tools to another's as represented in the diagram below.



XPDL

The eXtensible Process Definition Language (XPDL) is a file format standard designed for the exact purpose of exchanging process designs. It is actually a third-generation process language developed over the course of ten years by a collaboration of process vendors and users under the auspices of the Workflow Management Coalition (WfMC). The first version, released in 1998, was called WPDL. At about that time, XML became popular, and the second version, called XPDL 1.0 and released in 2002, used XML to express slightly improved semantics. The advent of Business Process Modeling Notation (BPMN) drove the third version, XPDL 2.0, which came out in October of 2005 and contained the capability to express all the semantics of BPMN. This evolution represents a continual refinement of proven steps forward. Full upward *and downward* compatibility has been maintained. It comes as no surprise that any product that can read XPDL 2 can by definition read XPDL 1, but perhaps less expected is that any product which reads XPDL 1 can also read XPDL 2 by simply ignoring the new tags. It is not an offering from a single vendor trying to gain an edge in the market, but a collaboration of numerous vendors and users, and the fact that it has stood the test of time is evidence that it has what is needed.

For a relatively unknown standard, a surprising number of products have implemented it. At last count, close to three dozen key process technology vendors already support XPDL. At the recent Shared Insights BPM conference in Boston (Nov. 2006), I verified that every vendor present, except two, supported XPDL. The official list is at the WfMC website, but current supporters include Adobe, Advantis, BEA (Fuego), EMC (Documentum Workflow), IBM (FileNet), IDS Scheer (Business Architect), Fujitsu (Interstage BPM), Global 360 (Capevisions, eiStream), Oracle9i (Warehouse Builder), Pega Systems, Software AG (Crossvision BPM), TIBCO (Staffware), Vignette (Process Workflow Modeler), and Zynium (Byzio), as well as many smaller vendors. What is really interesting is the number of open source initiatives based on XPDL: Enhydra JaWE (editor), Enhydra Shark (engine), Open Business Engine, and WfMOpen (engine). A couple of Visio-based process design tools support XPDL, including ITPCommerce and ITPearls, while Zynium offers Byzio, which can convert any Visio drawing to XPDL. XPDL is also used by simulation vendors (e.g., Simprocess). We are even seeing consulting firms standardizing on XPDL as they seek to increase their ability to reuse their collections of business process diagrams.

XPDL vs. BPEL

Whenever you begin to talk about process-oriented standards, the question always comes up about another process standard, Business Process Execution Language (BPEL, also known as WSBPEL and BPEL4WS). Many people automatically assume that BPEL and XPDL are direct competitors, but this is not at all true. BPEL and XPDL are entirely different standards for entirely different purposes. Many people today are disappointed by BPEL, but, in fact, BPEL does exactly what it set out to do. It was the amount of hype that led many people to believe it did much more.

BPEL is an "execution language," a programming language that has variables and operations. The operations can send and receive SOAP messages, and there is strong support for XML and XML transformation. It has constructs that make it easy to call multiple web services at the same time, and synchronize the results. It does not have any concepts to support the graphics of a diagram; activities do not have a position and size, and there is no need for a representation of an "arrow."

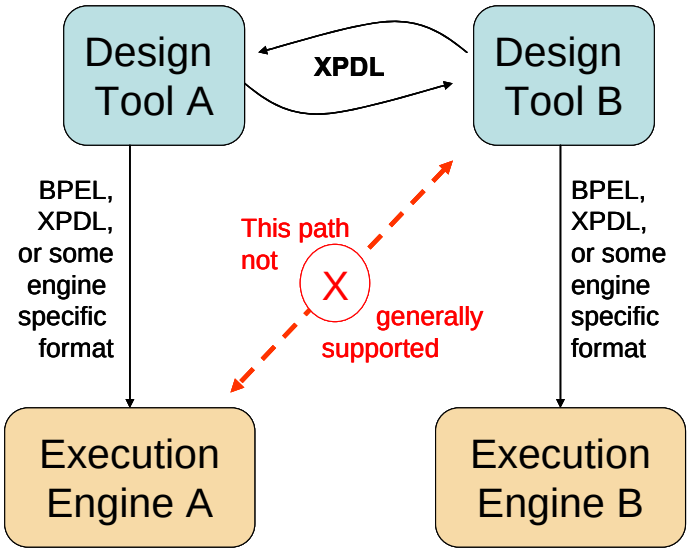
XPDL is a process design format, a file format that represents the "drawing" of the process definition. It has X & Y coordinates and node size. It has the concept of lines, and points along

the line that gives it a particular path. The nodes and lines have attributes that can specify executable information such as roles, activity descriptions, timers, web service calls, etc. XPDL 2.0 contains extensions in order to be able to represent all aspects of BPMN.

The goal of BPEL is to provide a definition of web service orchestration, the underlying sequence of interactions and the flow of data from point to point. Ultimately, BPEL is all about bits and bytes being moved from place to place and manipulated. It does not, however, attempt to represent the drawing that was used to specify the orchestration.

The goal of XPDL is to store and exchange the process diagram. It allows one process design tool to write out the diagram, and another to read the diagram, and for the picture to remain as similar as possible. It does not, however, guarantee the precise execution semantics. As you see, BPEL and XPDL are entirely different standards for entirely different purposes.

The different usage is best represented by the diagram below. At the top are various design-level tools. At the bottom are execution environments. As indicated, only XPDL can be used to carry the design from design tool to design tool. BPEL, XPDL, or other formats might be used to communicate the executable process to the engine. Generally, a vendor-specific design tool is necessary to translate the design into an engine-specific format. Generally, it is not possible to take executable code from one vendor's design tool, and execute it in another vendor's engine. Even BPEL, which many believe was created for this purpose, does not at this time allow different engines to run copies of each other's BPEL code (except in the simple cases).

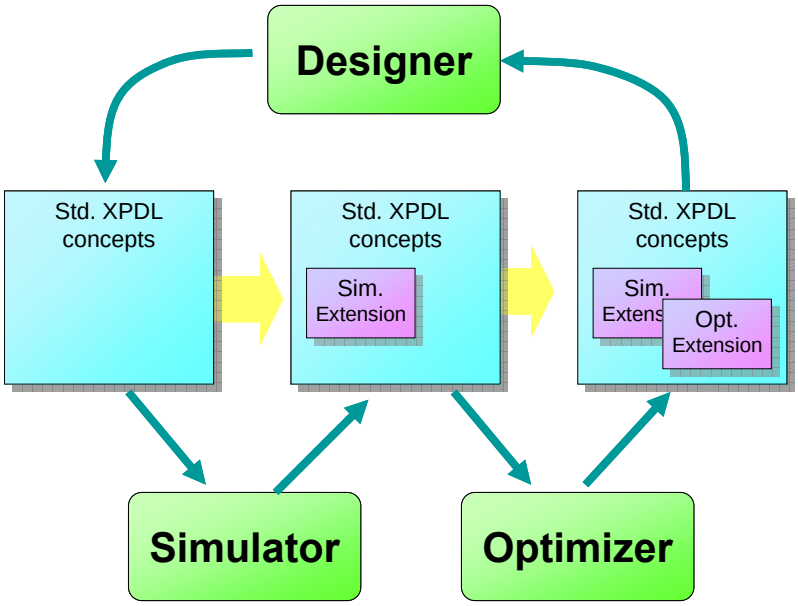


The XPDL file can provide this design interchange because it offers a one-for-one representation of the original BPMN process diagram. It can be written and re-read to recover the original diagram. BPEL, by contrast, is a non-trivial mapping, which is widely recognized as being one directional: You can take a BPMN diagram and produce BPEL, but it is difficult or impossible to recover the original BPMN diagram from the BPEL. This is not surprising since BPEL was not designed for process design interchange.

You might ask, if BPEL does not manage to communicate the execution representation to various engines with complete fidelity, why then would we expect XPDL to exchange the process diagram

with complete fidelity? The answer is simply that it does not need to. One design tool does not understand the output from another tool completely, but every design tool understands the most important parts (the form and shape of the diagram) as well as many standard BPM attributes. Because the model is communicated from design tool to design tool, if the transfer is not perfect, you have a chance in the receiving tool to fix it. Not perfect, but both useful and pragmatic.

Not every tool needs to understand the complete diagram. A simulation tool, for instance, will use the standard parts of the diagram, but probably ignore things like the attributes that define web service calls, since the simulation does not need to know this.



Extensibility

One of the most important aspects of XPDL is the extensibility mechanism. Each tool has specialized requirements of the diagram, and it can represent these using extended attributes. Other tools will not understand these extensions, but they will carry the extensions along. Thus, a tool specialized to clean up the layout might manipulate the graphical aspects of the model and return a clean model, including all the extensions, back to the original source without losing any information. Enhydra JaWE, for instance, is an open source XPDL editing tool that has been publicly demonstrated to read an XPDL file from Fujitsu's Interstage BPM, edit it, and return it without the loss of vendor-specific extensions. JaWE even allows users to view and modify the extended attribute values.

Some execution engines take XPDL directly. Fujitsu's Interstage BPM, Software AG's Crossvision BPM, and a number of open source engines do this because they are workflow-style BPM, which represent human activities in such a way that they retain their identity even while executing. That is the choice that a particular engine makes. Similarly, one can assume that an engine that does mainly web service calls would prefer BPEL.

What other standards exist beyond BPMN, XPDL, and BPEL? There are several other proposals for the communication of process definitions. A very interesting one is BPDM from OMG, which hopes to capture a common metamodel across all process definition formats. An initial proposal

for this was discussed at the OMG meeting in December 2006, and, while it appears to be promising, we must remember that it is merely a specification and is not built on field-proven standards. Beyond that, we are probably at least a year away from ratification of the specification, and at least two years away from vendor adoption – without any guarantee that it will ever be broadly adopted by multiple vendors. It will be interesting to watch the development of BPDM over the coming years, but the need for a Process Design Ecosystem, where different specialists within an organization can share their designs across different tools for different purposes, exists today.

One additional note: While process interchange is very important to customers who invest significant dollars in best-of-breed tools and a tremendous amount of time in developing their process diagrams, process *archiving* is also becoming very important. I was asked recently which file format should be used by a business that wants to preserve its investment and ensure stored processes would be readable in the future. The answer is clearly an XML format, and XPDL is the only standard XML file format supported today by dozens of vendors, and likely to be supported by tools in years to come. WfMC's commitment to upward and downward compatibility is the assurance that XPDL 3, whenever it comes out, will be fully compatible with the current versions.

Available today, XPDL is a proven format for process design interchange, and it is the most practical standard for establishing a Process Design Ecosystem.

Author

Keith Swenson can be contacted at KSwenson@usfujitsu.com

