The Business Process Modeling Notation (BPMN) was originally defined by the Business Process Management Initiative (BPMI.org) in order to address the need for a standard notation that business process experts could use to model business processes. Today, most business process management and enterprise architecture suites and tools support BPMN.

In June 2005, BPMI.org merged with the Object Management Group (OMG), a mature standards body best known in the enterprise computing world for its stewardship of the UML, XMI, MOF, and other Model Driven Architecture standards. The OMG is close to releasing its first official version of the BPMN standard, which will be publicly available for download and use.

**Requirements for BPMN 2.0**

As BPMN became entrenched in the market, inevitably its users identified improvements that could enhance its value. Some of the main issues with the BPMN specification are:

- It does not define a format for interchanging business process models among tools.
- It has no explicit metamodel for BPMN.
- It does not specify a means for interchanging diagrams among tools.
- The semantics of the various modeling constructs are not well defined.
- Its facilities for expressing choreographies are limited.

**Interchange Format**

A standardized interchange format enables a BPMN tool that complies with the standard to import a business process model that was created by another compliant BPMN tool. Today, some tools use the Workflow Management Coalition's XPDL as a BPMN interchange format, but XPDL is not truly based on BPMN's modeling constructs and thus the mapping to XPDL is not as direct as...
many would like.

Furthermore, companies who are investing in modeling infrastructure supported by the Eclipse Modeling Framework (EMF) and the OMG's Meta Object Facility (MOF) need an XML-based interchange format.

The BPMN specification does not provide an interchange format, nor does it codify any other interchange format as the official BPMN interchange format.

Metamodel

In order to define an interchange format that is well aligned with BPMN's modeling constructs, it is necessary to define a metamodel. One of the things that a metamodel does is to define precisely the information structures that a tool uses to capture the content of a business process model. Each modeling construct has a corresponding information structure that captures its content.

For example, BPMN defines a modeling construct called a Gateway. There are four kinds of gateways: Exclusive, Inclusive, Complex, and Parallel. There are two kinds of Exclusive gateways: Data-Based and Event-Based. A metamodel defines a schema of some sort that has the information structures needed to capture this information. It might define a Gateway structure that has a property that indicates which kind of Gateway is being captured; or it might define a separate structure for each kind of Gateway. These are different ways that the model content could be structured when it is stored in a repository.

When a modeling tool stores or transmits a business process model, it does so in accordance with the information structures that the metamodel defines.

The metamodel's definition of the structure of the model content is called the abstract syntax. Different concrete syntaxes, such as a relational schema or an XML schema, can be generated from the abstract syntax. EMF and MOF typically use a UML class model to express the abstract syntax of a language, and use the abstract syntax to automatically generate an XML-compatible XML schema that serves as the interchange format. The abstract syntax can also be the basis for generating Java interfaces and classes. EMF can also accept an XML schema as a representation of the abstract syntax.

For storage on disk, the tool may use a relational database with a relational schema that corresponds to the abstract syntax, while using an XML schema, derived from the same abstract syntax, as the format for interchange with other tools.

Today, BPMN does not define an explicit metamodel. Although one might argue that there is an implicit metamodel that the notation implies, in many cases the implications are not obvious.

Diagram Interchange

A diagram is a window for viewing a model's underlying content, and the same content can be portrayed differently by different diagrams. A diagram often suppresses and displays certain elements of a model in order to focus on a specific concern, and, in so doing, arranges the visual presentation in a certain way. Another diagram working off the same model content may suppress and display different parts of the model. For example, a diagram may show only part of a business process model, even though the underlying model captures much more than is visible in that diagram.

So it is not enough to simply exchange the underlying model content among tools. An interchange format also must define how to exchange model diagrams. It is also not sufficient to simply exchange jpg files or similar kinds of graphic files, because, to an importing tool, such a file is just an undifferentiated blob whose parts the tool cannot align with the underlying model content.

Therefore, an interchange format defines data structures that capture the relative positions of the different shapes and lines that appear in a diagram, and that capture the correspondences between the shapes and lines on the one hand and the underlying model content on the other.
Combining a diagram interchange format with an interchange format for exchanging the underlying model content yields a really complete interchange format. Once again, some tools use XPDL to exchange diagrams, but the BPMN specification is silent on the subject.

**Execution Semantics**

Although a complete, standardized interchange format allows tools to exchange and align the content and diagrams that constitute a business process model, standardization of the interchange format does not guarantee that the tools will interpret the meaning of the same model in the same way. When multiple companies want to combine parts of their business processes to form a business network, it helps if they can exchange their process models among the various business process management suites that they use, but if the suites do not agree on the semantics of those models, it increases the impedance encountered when using the models to support the execution of the business network.

Thus, the fact that BPMN's definition of semantics is neither complete nor rigorous creates head winds as companies pursue the increasing imperatives to form, optimize, and reconfigure business networks.

**Choreographies**

Many people in the business process management field use the terms *choreography* and *orchestration* to distinguish between B2B and internal business processes, respectively. The term "orchestration" originates from the idea that, in a business process management environment, a business process execution engine orchestrates internal processes by executing a business process model, whereas in a B2B scenario there can be no central orchestrator. "Orchestration" is a somewhat misleading term in that there are other modalities for executing internal business processes besides centralized orchestration, but we are stuck with this term nonetheless.

Models of B2B business processes hide the business partners' internal processes while exposing a common model of a choreographed exchange of messages among the partners.

BPMN today supports modeling orchestration and choreography via the concepts of swim lanes and pools, but overall its support for choreography is weaker than its support for orchestration. Its facilities are also rather limited for expressing how a company's internal processes relate to a B2B choreographies being used for commerce in business networks.

**The OMG’s RFP for BPMN 2.0**

The OMG issued an RFP for BPMN 2.0 in June 2007. ¹ It more or less covers the requirements outlined above.

The OMG’s standards development process is quite rigorous. ² An RFP kicks off the process of producing specifications. In order to be issued, an RFP must be approved by a majority vote of the administering Task Force, and then must pass a majority vote of the Architecture Board, then a majority vote of the general membership.

Specifications that are submitted in response to an RFP must be approved by a majority vote in the administering Task Force, then by a majority of the Architecture board, and then by a two-thirds supermajority of the general membership.

Specifications that clear these hurdles also require certification by the Board of Director’s Business Committee that there has been some commercial implementation, and, finally, approval

---

¹ The RFP is accessible to OMG members at http://www.omg.org/cgi-bin/doc?bmi/07-06-05

² For a detailed explanation of the OMG standards development process, see my blog at https://www.sdn.sap.com/irj/sdn/weblogs?blog=/pub/wlg/7096
by a majority of the full Board of Directors before the specification is available as a standard to
the general public.

The OMG’s Business Modeling and Integration Task Force has jurisdiction over BPMN 2.0. It
was formed following the merger of BPMI.org into the OMG.

**Initial Responses to the RFP**

Two teams of companies have submitted proposed specifications.\(^3\) Their contributions are called
“initial submissions,” so they are not yet complete, but they are substantive enough to make
evident the different approaches that they take.

One submission is called *BPMN for Services (BPMN-S)*. The other one does not have an official
name, so I’ll refer to it as the BEA-IBM-Oracle-SAP submission, since those companies constitute
the initial submission team.

In keeping with the RFP, the submissions have four aspects: metamodel, notation, interchange
format, and execution semantics, and those aspects cover choreography and orchestration
processes.

The biggest difference between the two submissions is in how they define the BPMN 2.0
metamodel. The BPMN-S submission positions the OMG’s Business Process Definition
Metamodel (BPDM) as the metamodel for BPMN 2.0. The BEA-IBM-Oracle-SAP submission
defines a dedicated BPMN 2.0 metamodel, and proposes a mapping between the dedicated
metamodel and BPDM.

In order to explain the differences in approach to defining the BPMN 2.0 metamodel, I should
provide some background on BPDM.

The OMG issued the BPDM RFP in 2003. The RFP opens with a statement of its objective:

> This Request For Proposals solicits submissions that specify a business process
definition metamodel, which is platform independent with respect to specific business
process definition languages. This metamodel will define an abstract language for
specification of executable business processes that execute within an enterprise…

After initial BPDM submissions came in to the OMG, BPDM languished with little to no activity for
quite some time. In 2006, activity picked up again, led by a new set of players, who are the same
parties that form the core of the BPMN-S submission team. BPDM is now going through the final
stages of completion. The BPDM specification includes a mapping to BPMN notation.

The BPMN-S submission uses BPDM as the metamodel, and uses BPDM’s mapping to BPMN
notation.

The BEA-IBM-Oracle-SAP submission takes the position that BPDM is not a metamodel of
BPMN; rather, it says, BPDM is a metamodel of a new, abstract language for process that, as
envisioned by the BPDM RFP, was intended to be mapped to multiple concrete languages. The
BEA-IBM-Oracle-SAP approach is that BPMN, as one of those concrete languages, requires its
own metamodel, whose constructs are clearly recognizable as BPMN elements. The submission
says, furthermore, that if BPDM is to play a role in the BPMN picture, it should be via a mapping
between its abstract language metamodel and a concrete BPMN language metamodel.

There are also some interesting differences in how the two submissions approach execution
semantics. The discussions regarding execution semantics can get quite technical. The highly
technical aspects are primarily of interest to BPM tool and process execution engine developers.

---

\(^3\) The proposals are available to OMG members at [http://www.omg.org/techprocess/meetings/schedule/BPMN_2.0_RFP.html](http://www.omg.org/techprocess/meetings/schedule/BPMN_2.0_RFP.html), under the “Initial Submission List” section heading.
The two submission teams are actively engaged in discussions about a possible merger. While these talks are going on, the parties are reluctant to disclose the content of these negotiations. During the public March 2007 OMG meeting in Washington, DC the two teams presented their submissions to the OMG’s Business Modeling and Integration Task Force. For the next two and a half days, the two teams met privately to explore the possibility of working together for one unified submission, and discussions have continued since then.

**Moving Quickly**

The process of defining BPMN 2.0 is moving quickly. The companies involved have brought the requisite expertise to the effort and intend to move expeditiously. In future installments of MDA Journal I will provide updates as BPMN 2.0 solidifies.

---

**David Frankel is Lead Standards Architect for Model-Driven Systems at SAP Labs. He has over 25 years of experience as a programmer, architect, and technical strategist. He is the author of the book, “Model-Driven Architecture®: Applying MDA® to Enterprise Computing.” He also is lead editor of the book “The MDA Journal.” He served several terms as a member of the Architecture Board of the Object Management Group (OMG), the body that manages the MDA standards, and he has co-authored a number of industry standards. Recently he has been publishing and speaking about the role of model-driven systems in enterprise SOA and has been promoting the Business Process Expert community at [www.sap.bpx.com](http://www.sap.bpx.com).**