

Business Process Execution Language for Web Service (BPEL)

Jan Mendling

Vienna University of Economics and Business Administration

Augasse 2-6, A-1090 Wien, Austria

jan.mendling@wu-wien.ac.at

Abstract: BPEL is gaining increasing attention as a potential standard for the definition of executable business processes based on web services. This paper gives an overview of standardization efforts in the area of business process execution, of the main concepts of BPEL, and of its support in practice.

1 Business Process Execution and Standardization

The standardization of business process management and workflow technology has been discussed for more than ten years (see e.g. [Hol04]). Several standardization bodies have proposed specifications for different aspects of business process management. The five bodies that have gained the most attention in this context are WfMC, OMG, BPMI, W3C, and OASIS. The *WfMC* (Workflow Management Coalition) has been the first organization to promote workflow standards. Its Workflow Reference Model distinguishes five interfaces of a workflow system [Hol94]. From WfMC's set of specifications¹ the XPDL standard for process definition (interface 1) is the most prominent (see [Wor05]). The *BPMI* (Business Process Management Initiative) started off in 2000 as an industry consortium to promote business process management standards. In 2002 BPMI published BPML [Ark02], an XML-based language for the specification of executable processes with web service interaction, and in 2004 BPMN [Whi04], a corresponding visual notation for business processes. The *OMG* (Object Management Group) first got involved with workflow technology as they accepted the Workflow Management Facility specification as a standard in 1998. In 2005 OMG and BPMI agreed to merge their business process related activities. As a consequence, BPMN is now an OMG standard. The *W3C* (World Wide Web Consortium) has published several standards for web service choreography. Choreography describes the interaction of distributed processes from a global point of view. WS-CDL [KBR⁺05] is the most recent specification in this area. It is meant to be utilized in conjunction with process definition languages that define the private implementation of processes. *OASIS* (Organization for the Advancement of Structured Information Standards) is an in-

¹see <http://www.wfmc.org/standards/docs/Stds.diagram.pdf> for an overview of WfMC workflow standards and associated documents.

dustry group that defines XML-based standards for web services and business integration. OASIS participates e.g. in the specification of the ebXML framework. For further details on business process related standards see [MzMP05].

Since 2003, OASIS is also responsible for the standardization of *BPEL*². The work on BPEL started with a merger of IBM's WSFL process definition specification with Microsoft's XLANG which resulted in the first version of BPEL [CGK⁺02]. In 2003 BEA, SAP, and Siebel joined in to extend BPEL to version 1.1 [ACD⁺03]. Currently, the second version of BPEL is in the final phase of standardization³.

2 Main Concepts of BPEL

BPEL is an XML-based language that models a business process as a composition from a set of elementary web services. A so-called BPEL engine is a dedicated software component that is able to execute BPEL process definitions. Each BPEL process can be accessed as a web service of the BPEL engine, too. The BPEL specification depends on the W3C standards WSDL for web service description, XML Schema for the definition of data structures, and XPath for retrieval of XML elements. Six of BPEL's most important concepts are briefly presented in the following, i.e., partner links, variables, correlation, basic activities, structured activities, and handlers. We will use the element names of BPEL 2.

- *Partner links*: A partner link provides a communication channel to remote web services which are used in the BPEL process. A respective partner link type must be defined first to specify the required and provided WSDL port types.
- *Variables*: Variables are used to store both message data of web service interactions and control data of the process. A variable must be declared in the header of a BPEL process by referencing a WSDL or an XML Schema data type.
- *Correlation*: As BPEL supports long-running business processes, there may be several process instances waiting for web service messages at a certain point of time. A correlation set specifies so-called properties, i.e. XPath statements to retrieve message parts that are unique for a specific process instance. According to a certain property value, like e.g. *ordernumber* = 1002006, a message is handed to the matching process instance.
- *Basic activities*: The basic steps of a BPEL process are performed by basic activities. There are activities to send and receive messages from web services (receive, invoke, reply), to change the content of variables (assign), to wait for a certain period or up to a certain point in time (wait), or to terminate the process (exit, formally called terminate)⁴. The second version of BPEL introduces an activity to check conformance to a Schema (validate) and the possibility to add proprietary activities

²The old acronym is *BPELAWS* (Business Process Execution Language for Web Services), the new one *WSBPEL* (Web Service Business Process Execution Language). The acronym *BPEL* can be used for both.

³The Committee Draft is available at <http://www.oasis-open.org/committees/download.php/16024/wsbpel-specification-draft-Dec-22-2005.htm>.

⁴Basic Activities in BPEL 2 are: receive, invoke, reply, assign, validate, empty, throw, rethrow, exit, wait, compensate, compensateScope, and extensionActivity.

(extensionActivity).

- *Structured activities*: The control flow of basic activities can be defined in two different styles: block-oriented or graph-based. Both styles can be mixed. Block-oriented control flow can be defined with structured activities. BPEL offers activities to specify parallel execution (flow), conditional branching based on data (if-else) or on receipt of a message (pick), sequential execution (sequence), and different loops (while, repeatUntil, forEach). Structured activities can be nested. Scopes are special structured activities. They mark-off the scope of local variables and handlers. Control flow can also be defined graph-based, but without introducing cycles, using so-called links. A link represents a synchronization between two activities.
- *Handlers*: BPEL provides handlers to deal with unexpected or exceptional situations. Event handlers wait for messages or time events. They can be used to specify deadlines on the process level. Fault handlers catch internal faults of the BPEL process. If the fault cannot be cured, the compensation handler can be triggered to undo the effects of already completed activities. Finally, the termination handler offers a mechanism to force a process to terminate, e.g. due to external faults.

Even though BPEL supports a rich set of primitives to specify executable processes, there are still some features missing towards full-fledged business process specification. The extension activity of BPEL 2 is a useful anchor point to fill these gaps. Currently, there are several BPEL extensions in progress of development, in particular BPELJ⁵ for Java inline code, BPEL4People⁶ for human worklists, and BPEL-SPE⁷ for sub-processes.

3 BPEL Support

Several major software vendors support the standardization of BPEL⁸. Several of these companies already provide BPEL support in their products. Furthermore, there are also open source implementations of BPEL including ActiveBPEL, bexee, MidOffice, and Twister. Therefore, it should be expected that BPEL will soon become not only a de-iure, but a de-facto standard for the definition of executable business processes.

For vendors, there are basically two options to align with BPEL, either to develop a BPEL engine and related tools from scratch or to implement BPEL import and export for existing systems. Oracle is one of the few vendors who offers a generic implementation of BPEL called Oracle BPEL Process Manager. Several other companies have chosen to extend their systems with import and export. In this case, the BPEL process has to be mapped to the object structure of the target system. The mapping between BPEL and graph-based control flow, in particular with unstructured loops, is an interesting challenge for both academia and practice. So-called transformation strategies [MLZ06] can serve as

⁵<ftp://www6.software.ibm.com/software/developer/library/ws-bpelj.pdf>

⁶<ftp://www6.software.ibm.com/software/developer/library/ws-bpel4people.pdf>

⁷<https://www.sdn.sap.com/irj/servlet/prt/portal/prtroot/docs/library/uuid/5cbf3ac6-0601-0010-25ae-ccb3dba1ef47>

⁸WSBPEL Technical committee participants at OASIS are Adobe, BEA, EDS, Hewlett-Packard, IBM, IONA, JBoss, Microsoft, NEC, Oracle, Sterling Commerce, Sun, Tibco, and webMethods

a blue print for that. Ouyang et al. propose a transformation of unstructured loops to event-condition-action rules that are implemented via BPEL event handlers [OvdADtH06]. Another option, but not in the general case, is to derive structured from unstructured loops [ZHB⁺06]. The simplest solution to this problem for the vendors is to prohibit the definition of unstructured cycles in their process design tool. Then all import and export transformations between BPEL and internal graph-based representation can be implemented without loss of information.

References

- [ACD⁺03] T. Andrews, F. Curbera, H. Dholakia, Y. Goland, J. Klein, F. Leymann, K. Liu, D. Roller, D. Smith, S. Thatte, I. Trickovic, and S. Weerawarana. Business Process Execution Language for Web Services, Version 1.1. Specification, BEA Systems, IBM Corp., Microsoft Corp., SAP AG, Siebel Systems, 2003.
- [Ark02] A. Arkin. Business Process Modeling Language (BPML). Spec., BPMI, 2002.
- [CGK⁺02] F. Curbera, Y. Goland, J. Klein, F. Leymann, D. Roller, S. Thatte, and S. Weerawarana. Business Process Execution Language for Web Services, Version 1.0. Specification, BEA Systems, IBM Corp., Microsoft Corp., 2002.
- [Hol94] David Hollingsworth. The Workflow Reference Model. TC00-1003 Issue 1.1, Workflow Management Coalition, 24 November 1994.
- [Hol04] David Hollingsworth. *The Workflow Handbook 2004*, chapter The Workflow Reference Model: 10 Years On, pages 295–312. Workflow Management Coalition, 2004.
- [KBR⁺05] Nickolaos Kavantzias, David Burdett, Greg Ritzinger, Tony Fletcher, Yves Lafon, and Charlton Barreto. Web Services Choreography Description Language Version 1.0. W3C Candidate Recommendation 9 November 2005, World Wide Web Consortium, April 2005.
- [MLZ06] J. Mendling, K. Lassen, and U. Zdun. Experiences in enhancing existing BPM Tools with BPEL Import and Export. In Schahram Dustdar et al., editor, *Proceedings of BPM 2006*, Lecture Notes in Computer Science, Vienna, Austria, 2006. Springer-Verlag.
- [MzMP05] J. Mendling, M. zur Muehlen, and A. Price. *Process Aware Information Systems: Bridging People and Software Through Process Technology*, chapter Standards for Workflow Definition and Execution, pages 281–316. Wiley Publishing, September 2005.
- [OvdADtH06] C. Ouyang, W.M.P. van der Aalst, M. Dumas, and A.H.M. ter Hofstede. Translating bpmn to bpeL. BPMCenter Report BPM-06-02, BPMcenter.org, 2006.
- [Whi04] S. A. White. Business Process Modeling Notation. Specification, BPMI.org, 2004.
- [Wor05] Workflow Management Coalition. Workflow Process Definition Interface – XML Process Definition Language. Document Number WFMC-TC-1025, October 3, 2005, Version 2.00, Workflow Management Coalition, 2005.
- [ZHB⁺06] Wei Zhao, Rainer Hauser, Kamal Bhattacharya, Barrett R. Bryant, and Fei Cao. Compiling business processes: untangling unstructured loops in irreducible flow graphs. *Int. Journal of Web and Grid Services*, 2(1):68–91, 2006.