A Reference Architecture for Semantic Business Process Management Systems[1]

Dimka Karastoyanova[1], Tammo van Lessen[1], Frank Leymann[1], Zhilei Ma[1], Jörg Nitzsche[1], Branimir Wetzstein[1], Sami Bhiri[2], Manfred Hauswirth[2], Maciej Zaremba[2]

[1]Institute of Architecture of Application Systems, University of Stuttgart, Germany
{firstname.lastname}@iaas.uni-stuttgart.de
[2]Digital Enterprise Research Institute, University of Ireland, Galway, Ireland
{firstname.lastname}@deri.org

Abstract: Semantic Business Process Management (SBPM) enhances BPM with semantic technologies in order to increase the degree of automation in the BPM lifecycle and help in bridging the gap between the business and IT views on business processes. In this paper, we describe the architecture of an SBPM System (SBPMS) which supports the whole SBPM lifecycle by providing functionality for process modeling, process configuration, process execution, and process analysis. We analyze the functional requirements of the SBPMS from the business user's and the IT expert's point of view and derive and describe the components of the SBPMS and their key interactions to achieve the required functionalities. We show how existing BPMS components can be extended to use semantics, and describe the integration of new components, such as a Semantic Execution Environment. The presented SBPMS is based on BPMN, BPEL and WSMO technologies.

1 Introduction

Business Process Management (BPM) encompasses methods, techniques, and tools that allow organizing, executing, and measuring the processes of an organization. Recently, in a "third wave" of BPM [SF03], business process management is increasingly supported by software components which address a specific part of BPM and which can be integrated into so-called BPM Systems (BPMS), thus bringing a flavor of flexibility and component-orientation into business process management. A BPMS is used by both business users and IT experts and supports the whole BPM lifecycle by providing functionalities for process modeling, process configuration, process execution, and process analysis. As an increasing number of functionalities in enterprises are automated, i.e., executed without the need of human intervention, a BPMS must support integration of and access to existing IT systems.

The ensuing integration problems can be addressed through the use of the Service Oriented Architecture (SOA) paradigm [Bu00]. SOA mandates the exposure of functionalities as services where the granularity of each service interface corresponds to

[1] The work presented in this paper was supported (in part) by the European project SUPER No FP6-026850.

a complete and meaningful business function. Thus, business processes can directly reference IT services. Such services are typically accessible via an Enterprise Service Bus [Ch04], which enables discovery and invocation of services.

Today's BPM software still requires a lot of human intervention throughout the BPM lifecycle. In particular, there are substantial difficulties and efforts incurred when it comes to bridging the gap between the business view and the IT view on business processes, for example, when a high-level business process model designed by a business user has to be translated to an executable process model by IT experts. These difficulties, often referred to as the Business-IT gap, are caused partly by the lack of understanding of the business needs by IT experts on the one hand and of technical details unknown to the business experts on the other hand. This results in significant delays between completion of the business view of the business process and the time it is ready for execution (long time to market).

The vision of Semantic Business Process Management (SBPM) is to close the Business-IT gap by using semantic technologies[2] [HLD+05]. Like Semantic Web Services (SWS) that achieve a higher degree of automation in discovery and mediation compared to conventional Web Services, SBPM attempts to improve the level of automation in process modeling, configuration, execution, monitoring, and analysis by using ontologies and Semantic Web Services technologies.

In this paper, we present the architecture of a Semantic Business Process Management System (SBPMS) that supports the whole SBPM lifecycle. In Section 2 we identify the functional requirements on the SBPMS from the business users' and the IT experts' point of views and explain why semantics is needed. In particular, we identify improved functionalities supporting business and technical experts, which result from the use of semantic technologies in BPM. In Section 3, we give a high level overview of the architecture of the SBPMS and explain our choices of base technologies. Based on the requirement analysis, in the following sections we describe the components that are involved in an SBPMS and how they interact to achieve the required functionality.

2 Functional Requirements on the SBPMS

The functional requirements are expressed in terms of the functionalities the SBPMS needs to provide in each phase of the SBPM lifecycle [Fa07, WMF+07]. In practice there are many definitions of phases in the SBPM lifecycle. In this paper we distinguish four phases during a SBPM lifecycle: Semantic Business Process (SBP) Modeling, SBP Configuration, SBP Execution, and SBP Analysis.

[2] In this paper we refer to Semantic Web and Semantic Web Services technologies as semantic technologies.

2.1 SBP Modeling

SBP Modeling is the first phase in the SBPM lifecycle. In this phase business users create and modify SBP models. Process modeling is usually done using a business process modeling tool, which allows business users to draw process elements, specify the process flow and additional relevant properties of a process model. The created SBP models are stored in a SBP repository. Besides these features, an SBP modeling tool should also allow business users to annotate process elements by referencing ontology entities, i.e. concepts, relations, axioms. The semantics of the tasks and the decisions in the process are thus no more specified just in natural language, but explicitly in a machine-readable manner. By utilizing semantic annotations and the resulting capability to perform automatic semantic-based discovery, the SBP modeling tool and the SBP repository provide advanced features that simplify SPB modeling, namely process fragment discovery, auto-completion, as well as creation of process and data mediators.

Business process models can be divided into semantically meaningful process fragments for later reuse. Therefore, the SBP modeling tool must allow business users to select a part of a process model and save it in the SBP repository as a process fragment. During process modeling business users require support for discovering existing process models or process fragments by the SBP modeling tool. If the discovery yields no results, the desired process models or process fragments should be composable from existing process artifacts. In order to improve simplicity of SBP modeling, the auto-completion functionality, typical for most integrated development environments, is also demanded in process modeling. Auto-completion facilitates searching automatically for process models or process fragments to complete an unfinished process part.

When modeling collaborative business processes based on existing process models, business users are frequently confronted with heterogeneity of data formats and behavior mismatches. Usually none of the parties that are involved in the collaborative business process changes their process model to resolve such mismatches. Therefore, the SBP modeling tool provides support for creating process mediators which mediate between two existing incompatible processes in order to make them interoperable.

2.2 SBP Configuration

The SBP Configuration phase encompasses all the necessary steps to make a process model executable. This includes appropriate specification of activity implementations[3] and creating (Semantic) Web Service descriptions of the services that the process provides. The usage of semantic technologies enables novel and more expressive ways of discovering services and binding them statically to activities or discovering the implementations of semantically described activities dynamically during runtime (“late binding”). In the latter case it should be assured at configuration time that for every activity in the process model there is at least one service that implements the required functionality.

[3] In this paper, we consider only automated activities in SBP Configuration, but no human tasks.

If the discovery of a Semantic Web Service (SWS) implementing a required functionality fails, the system can try to find a suitable composition of existing Semantic Web Services to achieve the required functionality. The composed SWS can again be either statically bound to an activity or dynamically during runtime. By using semantic technologies the discovery of activity implementations is automated. However, in some cases the generated executable processes have to be refined by IT experts. The refinement includes technical aspects like transaction boundaries and security aspects. The final step of the SBP Configuration is the deployment of the executable process model to the process engine.

2.3 SBP Execution

To accomplish a task in the process, the process engine invokes Semantic Web Services or conventional Web Services. A requirement imposed on SBPM systems is to support discovery, selection and invocation of services at runtime based on the semantic annotations of tasks of the deployed process. In case the interaction with the discovered service requires process and data mediation, the respective mediators defined in the modeling phase are utilized. All components that are involved in process execution must publish all relevant events for inclusion into the execution history.

2.4 SBP Analysis

SBP Analysis comprises process monitoring and process mining [APA+07]. Process monitoring addresses the need of business users to monitor the key performance indicators that have been defined during process modeling and of IT experts to monitor the execution status of running process instances. In general, process mining operates on the execution histories of finished processes that are logged during the SBP Execution phase. Process mining algorithms try to discover explicit process models based on execution histories for conformance checking and optimization of process models. However, most execution histories contain only technical information about completed process instances, such as processing and wait time, involved resources, and frequencies. As in SBPM the events are semantically annotated, a meaningful link to business information is created through these semantic annotations, such as which business object was being worked on, what is the business semantics of the last event finished etc. Based on the semantic annotation of events, reasoning mechanisms can be employed to enable richer monitoring and querying of events.

3 SBPMS Architecture

Based on the requirements described in the previous sections we have defined an SBPMS architecture to address them. We first give a high level description of the architecture, introduce the technologies used, and then describe the architectural components and their dependencies (Sections 4-8).

Figure 2 shows the high-level view of the SBPMS architecture. The architecture is composed around a Semantic Service Bus (SSB) [KLN+07], which provides the central communication infrastructure for all SBPMS components. Furthermore, the SSB provides functionality for deployment of business processes, enables invocation of Web Services via a variety of protocols, and provides an infrastructure for publishing of events for monitoring purposes. The SBP Execution Engine (SBPEE) [LND+07] is in charge of executing semantic business processes. It delegates the discovery and invocation of services to the SSB which uses the Semantic Execution Environment (SEE) for the discovery and selection of Semantic Web Services.

The SBP Modeling Tool [DSS+07] supports process modeling and configuration; the SBP Monitoring & Management Tool and the SBP Mining Tool encapsulate the monitoring and mining functionalities described in the previous section. All components can use a set of "support" components, which comprise repositories for storing semantic business processes, ontologies, SWS and the execution history of the whole execution infrastructure (SBPEE and SEE), and components for lifting and lowering of data, mediation, composition, process discovery and reasoning. All these components are also available as services on the SSB.

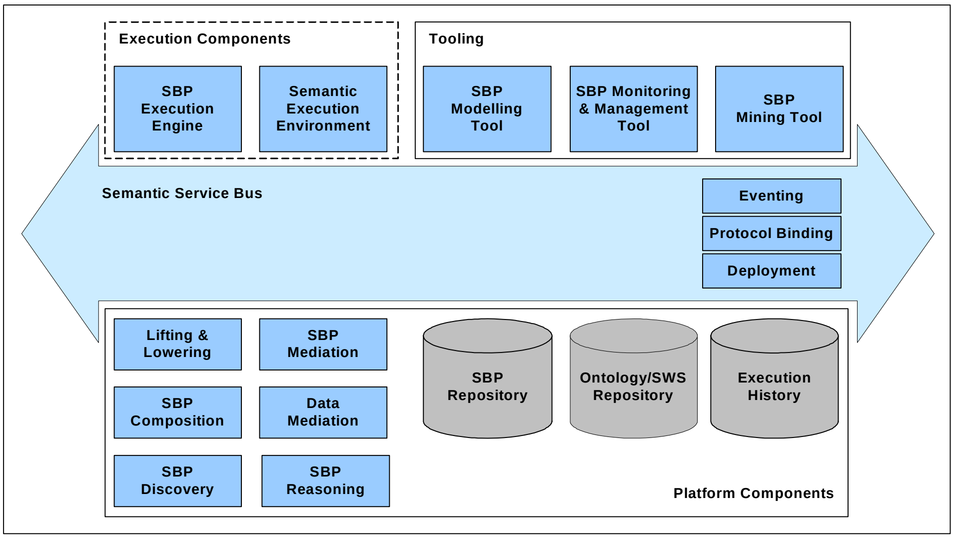

Figure 1: SBPMS Architecture

This conceptual SBPMS architecture can be realized through various implementation technologies. We have chosen the following base technologies:

- The SBPMS architecture is utilizes the Web Service technology [WCL+05].
- BPEL [BPEL07] is used as the standard process execution language.
- Because of its wide acceptance we use BPMN [BPMN06] as the process modeling notation.

- We use WSMO [RLK+06] as the SWS framework.

4 Ontological Framework and Reasoning

The use of ontologies is a key concept that distinguishes SBPM from conventional BPM. Within SBPM two types of ontologies are used: domain ontologies and SBPM specific ontologies. Domain ontologies support process modeling in terms of describing the actual data that is processed during process execution. Via this semantic description of the data, business process analysis can be semantically enhanced since the semantic meaning of the data is preserved during all phases of the process lifecycle.

The SBPM specific ontologies [HR07] support the different phases of the lifecycles and different views on processes. We have defined (i) an ontology for BPMN, (ii) sBPEL, an ontology for BPEL4SWS [NLK+07], which is an extension of BPEL enabling orchestration of Semantic Web Services, (iii) an event ontology that semantically describes the audit trail data of the whole SBPMS, and (iv) the Business Process Modeling Ontology (BPMO).

BPMO is in the very center of the process ontologies stack. It provides a notation-independent view on business processes and simplifies the translation of modeling centric notations (BPMN) into executable process models (BPEL4SWS) and vice versa. Thus, it helps in bridging the gap between the business view on processes provided by the modeling notations and the IT view on processes. Using appropriate ontology mediation techniques, a process model can be viewed from different perspectives enabled by the different notations.

All SBPM ontologies are expressed in WSML Flight [WSML05] and the SBP Reasoner component provides an inference engine for reasoning tasks on these ontologies. WSML Flight is also used as the query language in all the repositories. Thus the process reasoner can access the process-related artifacts and derive implicit knowledge.

5 Semantic Service Bus

The Semantic Service Bus (SSB) [KLN+07] provides serves as the integration middleware in the SBPMS architecture. In addition, the SSB enables deployment and monitoring of business processes. The SSB supports a wide range of standard access mechanisms and protocols. Internally, the SSB uses normalized messages that encapsulate the actual message information and hide protocol specific details [Ch04].

In our architecture all component interfaces are described in WSDL and we distinguish two types of service components: components that expose their functionality as a service on the bus, and components that provide a platform for hosting services, so called service containers. A service container can itself be exposed as a service on the bus. The SBPEE, for instance, hosts several BPEL processes which are externalized as services on the bus. Additionally, it is itself exposed as services, i.e., providing functionality for process deployment. To make the services accessible to the outside world, they communicate via protocol binding components that transform internal, normalized messages into external, protocol-specific messages.

The SSB also provides a deployment component. For deployment, the user defines a Semantic Process Artifact Bundle (SPAB) which contains all necessary artifacts for multiple service engines or protocol binding components. The deployment component distributes these artifacts to the components in charge and ensures atomic semantics for the deployment of semantic business processes.

The SSB also includes an event infrastructure which allows the SBPMS components to publish and consume logging events. Publishing components are for instance the SBPEE and the SEE. Examples of consuming components are the SBP monitoring and mining tools.

6 Process Modeling Environment

The process modeling environment contains components that enable process modeling and configuration. It comprises a graphical SBP modeling tool, an SBP composer, an SBP discoverer, and a SBP repository [MWA+07], and uses an ontology and SWS repository.

The SBP modeling tool supports process modeling based on the business process modeling ontology (BPMO) visualized using BPMN. The process models are annotated using several ontologies, such as organizational ontologies and domain-specific ontologies [HR07]. The SBP modeling tool provides functionality for process fragment discovery and auto-completion by using the SBP discoverer. The modeling tool sends a process query to the SBP discoverer, which retrieves the process fragments from the SBP repository. As the process fragments and process models are semantically annotated, the discovery procedure can use semantic matchmaking techniques thus improving the quality of machine-processed query results in comparison to non-semantic approaches.

When the BPMO process model is to be translated to an executable representation, the modeling tool may use the SBP composer which ensures that activities in the BPMO process have implementations in the form of SWS (modeled in WSMO). For each activity in the BPMO process model that is associated with a WSMO Goal (a semantic description of the functionality which is to be achieved by an SWS or a set of SWS), the process composer searches for existing SWS in the SWS Repository. If no matching SWS is found, the composer tries to “synthesize” a process fragment, using a set of AI planning techniques [WMD+07], that provides the required functionality by combining existing functionalities. The composed process fragment then contains only WSMO Goals which can be resolved at runtime into an SWS. The BPMO process is finally translated into an sBPEL process model and serialized into BPEL4SWS. In addition, corresponding WSDL and WSMO descriptions are generated, as each BPEL4SWS process model is itself a Semantic Web Service as well as a WSDL Web Service.

To resolve mismatches between interacting processes, data and process mediators are needed. These mediators can be modeled during design time using the SBP modeling tool, and are deployed to the SBP mediation component.

All modeled processes are stored in the SBP repository as instances of the BPMO ontology and sBPEL ontology. The SBP repository API provides interfaces for storing, retrieving and querying process artifacts, using WSML Flight logical expressions as the query language. The SWS/Ontology repository, in turn, stores artifacts related to SWS, i.e., goals, ontologies, mediators and semantic Web Services. At runtime, it serves also as a service registry.

7 Process Execution Environment

The process execution environment consists of the SBPEE and the SEE. Based on a BPEL 2.0 compliant process engine, the SBPEE is responsible for the execution of BPEL4SWS processes. BPEL4SWS [NLK+07] is an extension of BPEL 2.0 and enables orchestration of Semantic Web Services (SWS). To achieve a goal-based SWS invocation, the SBPEE passes a WSMO goal to the SEE, which dynamically discovers an appropriate Web Service and delegates its invocation to the SSB.

Before a semantic business process can be executed, it has to be deployed on the engine. This initial step is started by the SBP modeling tool by sending a SPAB to the Semantic Service Bus. The SPAB bundle contains the process model description and the interface descriptions (both conventional WSDL and WSMO SWS descriptions) of the process model and of the partner services. The SSB unpacks the SPAB and distributes the contained artifacts to the responsible components – the process model to the SBPEE, WSMO descriptions to the SEE and WSDL descriptions to the protocol binding components in order to enable the communication with conventional Web Services.

After deploying the process model, the SBPEE listens for incoming messages from the SSB, which are targeted to this particular process. By analyzing these messages, the engine decides whether the message belongs to a running process instance (correlation) or creates a new instance of the process model (instantiation). The SBPEE then navigates through the BPEL4SWS process, as specified by the control flow of the process model. For the invocation of WSDL Web Services, the SBPEE uses the protocol binding components of the Semantic Service Bus. When dealing with WSMO SWS, the SBPEE uses the SEE [ZMH+07] for SWS discovery, selection and data mediation. This means that after passing a WSMO Goal to the SEE, the SEE discovers an appropriate SWS that achieves that goal. The invocation may involve mediation functionality. WSMX [ZMH05] and IRS-III [DCH+07] are two implementations of the SEE.

Figure 3 depicts the interaction between SEE, the SSB and the SBPEE in the case of a goal-based SWS invocation. First a service requester sends a WSMO Goal, which semantically describes the functionality to be performed, to the SEE. The service requester can be an arbitrary Web Service client including a BPEL4SWS process. The SEE performs goal-based SWS discovery, which can use multiple discovery engines.

The discovery process takes into account both functional and non-functional (QoS) requirements. After an SWS is selected, the SEE creates the message which is to be sent to the service, and delegates the invocation to the SSB. In the case of an external Web Service, the actual invocation of the SWS is performed by a protocol binding component of the SSB depending on the protocol used by the SWS. If the SWS is hosted by a service container plugged into the SSB, the SWS is invoked via normalized messages. The SWS invoked may be any Web Service, even another BPEL4SWS process hosted by the SBPEE.

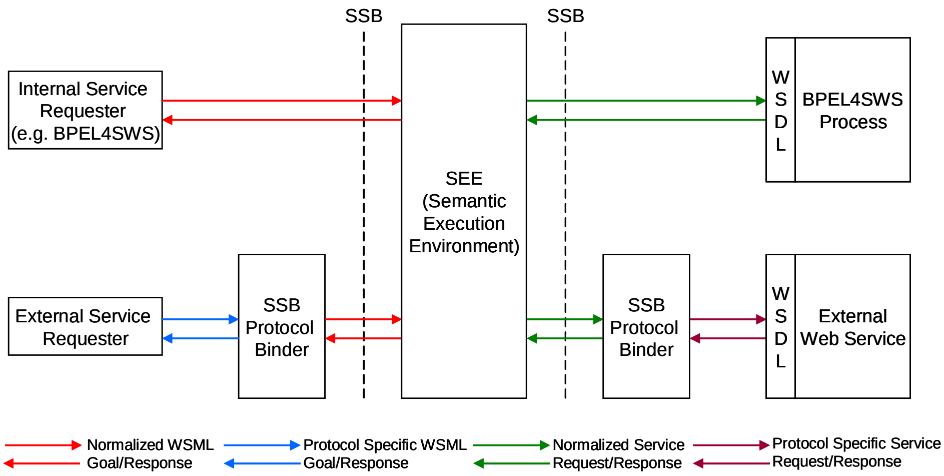

Figure 3: Goal-based interaction within the process execution environment

The invocation of the SWS may involve data mediation if service requester and service provider use different ontologies for describing their interfaces and data exchanged. Data mediation is provided by the SEE as a service. Process mediators are provided by the SBP mediation component and are exposed as SWS.

The SSB also provides lifting and lowering services which are needed to transform between the ontological and XML representation of data. Assume a client wants to invoke an SWS, e.g. a WSMO service. Therefore a goal including ontological instances is submitted to the SEE. After discovery the actual service has to be invoked. If the discovered service is a pure SWS, the communication can be performed on a semantic level, using ontological instances as instance data. However, if the service to be invoked is described as an SWS but in fact is a conventional Web Service, it has to be invoked using its WSDL interface. Thus, the semantic data has to be lowered to the XML representation the service can consume and XML data resulting from the invocation again has to be lifted to the semantic level to be sent back to the requester.

8 Process Monitoring and Analysis Environment

The process monitoring and analysis environment comprises the SBP monitoring & management tool, the SBP mining tool, and the execution history, which provides the input data for these tools. The basis for process monitoring and process mining are events which document the process execution. Each component sends its logging information to an event sink in the communication infrastructure, which then notifies other components interested in these events (the publish/subscribe paradigm is utilized).

In order to support the analysis of business processes, it is necessary to store all events in a globally shared persistent storage. This functionality is provided by the execution history component, which consumes all logging events from all components involved in an execution. In SBPM, all events are ontologically described and preserve links to both the components involved and to the particular invocation contexts that link events belonging to a certain execution. All events are described as instances of an Event Ontology.

The SBP monitoring & management tool supports passive and active monitoring. For active monitoring no human intervention is required, the monitoring tool simply gathers all specified information. In contrast to active monitoring, passive monitoring includes human intervention and allows business users to specify in an ad-hoc way the information to be provided for evaluating and analyzing running processes. The information to be provided is specified as queries against the execution history and can go far beyond active monitoring functionalities that only get the most essential information about the execution of processes. Since in an SBPMS, events (and thereby the execution history) are semantically described, the querying capabilities are more powerful compared to conventional BPMS.

To enable monitoring and management, the process execution environment components provide two kinds of information provisioning mechanisms: (1) During the execution, logging events are generated for each processing step. These events are consumed by the Execution History and the Monitoring Tool and can be visualized to show the current status of the execution. (2) The components provide interfaces for active monitoring. This allows external tools to ask explicitly for any information, e.g., for a list of process models deployed to the process engine or a list of completed process instances. Additionally it is possible to manage the lifecycle of particular services, e.g., start/stop/resume.

The SBP analysis tool uses process mining on the audit log data of the Execution History to analyze semantic business processes. The goal is to find aspects of the process model that can be optimized in terms of throughput, efficiency, etc., or to discover deviations between process models and their actual execution. In contrast to traditional process mining technologies, Semantic Process Mining benefits from the additional information in the logged audit data, i.e., the semantic annotations and classifications of instance data. This enables business users to perform more powerful analyses, e.g., to analyze all process instances that are related to a super class (e.g., “trading: fruits”), even if this class is not explicitly mentioned in a single process (“trading: apples”).

9 Conclusion

In this paper we presented the architecture of a Semantic Business Process Management System. We described the components, their functionalities and roles in the modeling, configuration, execution, and analysis phases of the SBPM lifecycle, and their dynamic collaboration. Our architecture relies on semantic enhancements of key BPM technologies like BPEL and BPMN. We use WSMO as the ontological framework for Semantic Web Services. The ontological descriptions of data and processes ensure a common view on Semantic Business Processes. They provide a logical foundation for machine-based reasoning that helps additionally in bridging the gap between the different representations of a process model on business and IT level. In addition, the use of ontologies enables improvements in BPM functionalities such as automated process fragment discovery and composition.

The proposed architecture supports the augmentation of BPM with semantics, i.e. it allows business users to add meaningful and machine-processable information, resulting in benefits throughout the entire BPM lifecycle: during the design time semantics provides improved support for process and service discovery and thus increases the reusability of process artifacts. Using semantics at run time introduces increased flexibility for service discovery and data mediation. The use of semantics during the analysis phase provides the basis for new types of process analysis based on semantically enriched process audit trails.

References[4]

[APA+07] Alves de Medeiros, A.K.; Pedrinaci, C.; Aalst, W.M.P.; Domingue, J.; Song, M.; Rozinat, A.; Norton, B.; Cabral, L.: An Outlook on Semantic Business Process Mining and Monitoring. In Proceedings of 3rd International IFIP Workshop On Semantic Web & Web Semantics (SWWS '07) at On The Move Federated Conferences and Workshops. (To appear)

[BPEL07] OASIS WS-BPEL TC: Web Services Business Process Execution Language Version 2.0. OASIS Standard. 2007. http://docs.oasis-open.org/wsbpel/2.0/OS/wsbpel-v2.0-OS.html

[BPMN06] Business Process Modeling Notation Specification. OMG Final Adopted Specification, February 6, 2006.

[Bu00] Burbeck, S.: The Tao of e-business services. 2000. http://www-128.ibm.com/developerworks/webservices/library/ws-tao/

[Ch04] Chappell, D. A.: Enterprise Service Bus. O'Reilly, 2004.

[DCH+07] Domingue, J.; Cabral, L.; Hakimpour F.; Sell, D.; Motta, E.: IRS-III: A Platform and Infrastructure for Creating WSMO-based Semantic Web Services. In Proceedings of the Workshop on WSMO Implementations (WIW2004), CEUR Workshop Proceedings, September 29-30 2004.

[DSS+07] Dimitrov, M.; Simov, A.; Stein, S.; Konstantinov, M.: A BPMO Based Semantic Business Process Modelling Environment, Proceedings of the

[4] All links used in this paper have been followed at November 26, 2007.

Workshop on Semantic Business Process and Product Lifecycle Management (SBPM-2007), Vol-251, CEUR-WS, June 2007, ISSN 1613-0073

[Fa07] Fantini, P. (ed.): Semantic Business Process Lifecycle. SUPER Project Deliverable D2.2. May 2007. http://www.ip-super.org/res/Deliverables/M12/D2.2.pdf

[HLD+05] Hepp, M.; Leymann, F.; Domingue, J.; Wahler, A.; Fensel, D.: Semantic Business Process Management: A Vision Towards Using Semantic Web Services for Business Process Management. Proceedings of the IEEE ICEBE 2005, October 18-20, Beijing, China, pp. 535-540

[HR07] Hepp, Martin; Roman, Dumitru: An Ontology Framework for Semantic Business Process Management, Proceedings of Wirtschaftsinformatik 2007, February 8 - March 2, 2007, Karlsruhe

[KLN+07] Karastoyanova, D.; Lessen, T. van; Nitzsche, J.; Wetzstein, B.; Wutke, D.; Leymann, F.: Semantic Service Bus: Architecture and Implementation of a Next Generation Middleware. 2nd International Workshop on Services Engineering (SEIW) 2007. Istanbul, Turkey, April 16, 2007.

[LND+07] van Lessen, T.; Nitzsche, J.; Dimitrov, M.; Konstantinov, M.; Karastoyanova, D. Cekov, L; Leymann, F.: An Execution Engine for Semantic Business Processes, SeMSoC-07: Proceedings of the 2nd International Workshop on Business Oriented Aspects concerning Semantics and Methodologies in Service-oriented Computing, September 2007

[MWA+07] Ma, Z.; Wetzstein, B.; Anicic, D.; Heymans, S.; Leymann. F.: Semantic Business Process Repository, Proceedings of the Workshop on Semantic Business Process and Product Lifecycle Management (SBPM-2007), Vol-251, CEUR-WS, June 2007, ISSN 1613-0073

[NLK+07] Nitzsche, J., van Lessen, T., Karastoyanova, D., Leymann, F.: BPEL for Semantic Web Services. In Proceedings of the 3rd International Workshop on Agents and Web Services in Distributed Environments (AWeSome'07), 2007. (To appear)

[RLK+06] Roman, D.; Lausen, H.; Keller, U.; et al.: D2v1.3 Web Service Modeling Ontology (WSMO). WSMO Final Draft 21 October 2006. http://www.wsmo.org/TR/d2/v1.3/

[SF03] Smith, H.; Fingar, P.: Business Process Management. The Third Wave. Meghan-Kiffer,US 2003.

[WCL+05] Weerawarana, S.; Curbera, F.; Leymann, F.; Storey, T.; Ferguson, D.: Web Services Platform Architecture: Soap, WSDL, WS-Policy, WS-Addressing, WS-BPEL, WS-Reliable Messaging and More. Prentice Hall PTR, 2005.

[WMD+07] Weber, I.; Markovic, I.; Drumm, C.: A Conceptual Framework for Composition in Business Process Management, BIS 2007: Proceedings of the 10th International Conference on Business Information Systems, 2007

[WMF+07] Wetzstein, B.; Ma, Z.; Filipowska, A.; Kaczmarek, M.; Bhiri, S.; Losada, S.; Lopez-Cob, J.M.; Cicurel, L.: Semantic Business Process Management: A Lifecycle Based Requirements Analysis, Proceedings of the Workshop on Semantic Business Process and Product Lifecycle Management (SBPM-2007), Vol-251, CEUR-WS, June 2007, ISSN 1613-0073

[WSML05] de Bruijn, J.; Lausen, H.; Krummenacher, R.; Polleres, A.; Predoiu, L.; Kifer, M.; Fensel, D.: The Web Service Modeling Language WSML. 5 October 2005.

[ZMH05] Zaremba, M.; Moran. M.; Haselwanter, T.: WSMX Architecture. WSMO Deliverable, 2005

[ZMH+07] Zaremba, M; Matthew, M.; Haselwanter, T.; Norton, B. : Semantic Web Services Architecture and Information Model, 2007 OASIS SEE TC Draft