Semantic Business Process Management: A Lifecycle Based Requirements Analysis

Branimir Wetzstein[1], Zhilei Ma[1], Agata Filipowska[2], Monika Kaczmarek[2], Sami Bhiri[3], Silvestre Losada[4], Jose-Manuel Lopez-Cobo[4], Laurent Cicurel[4]

[1]Institute of Architecture of Application Systems, University of Stuttgart, Germany
{branimir.wetzstein, zhilei.ma}@iaas.uni-stuttgart.de
[2]Department of Management Information Systems, Poznan University of Economics, Poland
{a.filipowska, m.kaczmarek}@kie.ae.poznan.pl
[3]Digital Enterprise Research Institute, National University of Ireland, Galway, Ireland
sami.bhiri@deri.org
[4]Intelligent Software Components S.A.
{slosada,ozelin,lcicurel}@isoco.com

Abstract. Despite of increasing software support for Business Process Management (BPM), currently there is still a low degree of automation in the BPM lifecycle, especially when it comes to bridge between the business and IT view on business processes. The goal of Semantic Business Process Management is to achieve more automation in BPM by using semantic technologies. In this paper, we describe on a conceptual level how ontologies and semantic web service technologies can be used throughout the BPM lifecycle, consisting of process modeling, implementation, execution, and analysis phases. The use of semantics in BPM results in new functionality a Semantic Business Process Management System (SBPMS) has to implement. For each phase of the BPM lifecycle, we identify the new functional requirements for a SBPMS, and explain the benefits of adopting semantic technologies in SBPM.

Keywords: Business Process Management (BPM), Semantic Business Process Management (SBPM), Semantic Web Services, Ontologies

1 Introduction

Business Process Management (BPM) is a top-down methodology designed to organize, manage, analyze, and reengineer the processes running in an organization. In the last few years, with the upcoming of the “third wave” of BPM [SF03], the BPM lifecycle has been increasingly supported by a set of software technologies, which have been bundled together to a so called BPM System (BPMS). A BPMS is used by both business people and IT engineers, and supports modeling, execution and monitoring of business processes in a unified manner. Typically, the BPM lifecycle begins with the business analyst creating process models using a modeling tool. In the next step the process model is translated by IT engineers to a workflow model, which is run on a process engine. The process engine executes the workflow model by

In: M. Hepp, K. Hinkelmann, D. Karagiannis, R. Klein, N. Stojanovic (eds.): Semantic Business Process and Product Lifecycle Management. Proceedings of the Workshop SBPM 2007, Innsbruck, April 7, 2007, CEUR Workshop Proceedings, ISSN 1613-0073, online CEUR-WS.org/Vol-251/

delegating the process tasks to human workers or automated IT applications. Finally, monitoring tools enable business analysts to measure the process performance.

Despite of increasing software support for BPM, there is still a low degree of automation in the BPM lifecycle. In particular, there are substantial difficulties when it comes to bridge the gap between the business and IT views on the business processes. One of the major problems is the translation of the high-level business process models, which are modeled by a business analyst, to workflow models, which are executable IT representations of the business processes. These difficulties, which result in time delays between design and execution phases of the process, and are caused partly by the lack of understanding of the business needs by IT experts and of technical details by business experts, are often referred to as the Business-IT gap.

The vision of Semantic Business Process Management (SBPM) is to close the Business-IT gap by using semantic technologies [HLD+05]. Similarly to how Semantic Web Services achieve more automation in discovery and mediation as compared to conventional Web services, in SBPM more automation should be achieved in process modeling, implementation, execution and monitoring phases by using ontologies and semantic web services technologies.

In this paper, we present our view on how semantic technologies can enhance BPM throughout its lifecycle. For each of the four phases, namely process modeling, implementation, execution, and analysis, we describe how semantic technologies can be used and depict the benefits of their usage. We identify new functionalities, which exploit the usage of semantics and which should be implemented by a Semantic Business Process Management System (SBPMS). We describe the functionalities a SBPMS should provide from a requirements perspective and do not show how these functionalities could be concretely realized, which is part of our ongoing and future work. Therefore, our description is mostly technology-independent.

The rest of the paper is organized as follows: section 2 gives an introduction into the BPM lifecycle. Section 3 then analyzes the requirements on the SBPMS for each phase of the BPM lifecycle. In section 4, a conclusion and an outlook are provided.

2 Business Process Management Lifecycle

In the following, we will describe the BPM lifecycle as supported by current BPM systems. This BPM lifecycle will serve as the basis for our discussion on SBPM requirements in the following chapter.

In the literature there is no uniform view on the number of phases in the BPM lifecycle. It varies depending on the chosen granularity for identifying the phases. In this paper we consider the following phases: process modeling, process implementation, process execution, and process analysis. We distinguish two roles in the lifecycle: business analysts or business managers, who create process models and analyze process models from the business point of view, and IT engineers, who are involved in process implementation and execution phases.

– **Process Modeling:** Process Modeling is the first phase in the BPM lifecycle. In this phase a business analyst creates an analytical process model with help of a modeling tool by specifying the order of tasks in the business process. The modeling tool typically supports a graph-based modeling approach adopting a popular process modeling notation such as Business Process Modeling Notation (BPMN) [BPMN06]. In addition to predefined graphical notations, business analysts have normally the possibility to specify some additional information in natural language for each element in a process model, such as what the tasks in the process are supposed to do and by whom they are expected to be performed. Process models created in the process modeling phase are usually too high level to be executed by a process engine because of lack of technical information such as binding of IT services and data formats for each task. Therefore, an analytical process model must be transformed to an executable process model, which is the focus of the process implementation phase.

- **Process Implementation:** In the process implementation phase a process model created in the process modeling phase is transformed and enriched by IT engineers into a process model, which can be executed in a process engine [LR00]. The standard language for describing executable processes in the context of Service-Oriented Architecture (SOA) and Web services [WCL+05] is the Business Process Execution Language (BPEL) [BPEL07]. The executable process model can only be partly generated from the analytical process model. The web services that are needed to execute the process model have to be manually and statically assigned. The same holds for data formats and data flow. The resulting executable process model can be deployed into a process engine for execution.
- **Process Execution:** After process deployment, the process engine executes process instances by navigating through the control flow of the process model. The process engine delegates automated tasks to Web services and manual tasks to human workers. In the context of SOA, the process itself is exposed as a Web service and can be invoked by other processes or other clients.
- **Process Analysis:** Process analysis comprises monitoring of running process instances and process mining. Process monitoring displays information on the running process instances, such as e.g. which branches of the control flow of a running process were taken; where in the control flow the process has halted after a failure; the current variable values of a process instance, etc. Some BPMSs support also business-level monitoring, where the business analyst can specify key performance indicators of the process during process modeling, and then gets them evaluated and presented in form of dashboards during process execution. The goal of process mining is to provide information necessary for potential optimization of the process model by using process mining algorithms [ADH+03]. Process mining operates on event logs, which are produced by the process engine during process instance execution, to analyze a set of finished process instances. Process mining algorithms deduce from the event logs how the process is in reality executed. The deduced process model can then be compared with the deployed process model and thus be used for conformance checking and optimization purposes. Process mining algorithms can also be used for performance analysis of processes.

3 Requirements Analysis for SBPM

The goal of SBPM is to combine BPM with semantic technologies, in particular ontologies and semantic web services (SWS), in order to achieve more automation in the BPM lifecycle and to provide more convenient features to business users and IT engineers. The usage of semantic technologies doesn't affect the main phases of the BPM lifecycle, but increases the automation degree within the phases and adds new or enhances existing BPMS functionalities. The SBPM lifecycle thus contains the following phases: SBP Modeling, SBP Implementation, SBP Execution, and SBP Analysis. Figure 1 depicts the SBPM lifecycle and the functionalities which are related to SBPM. In the following, we will describe the functional requirements for each phase of the SBPM lifecycle.

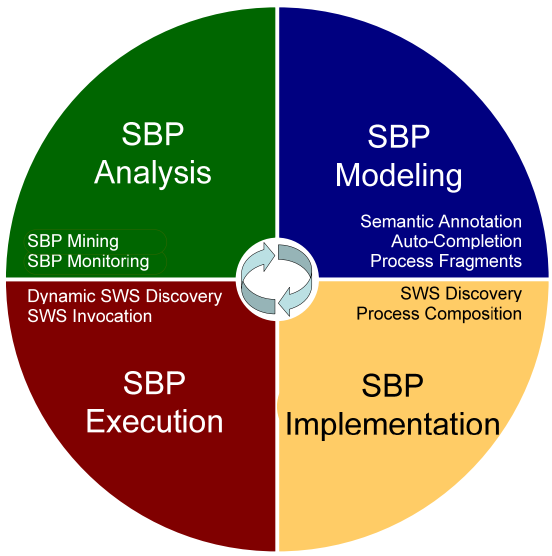

Figure 1: SBPM Lifecycle

3.1 Semantic Business Process Modeling

Semantic Business Process Modeling is the first phase of the SBPM lifecycle. It produces semantically annotated business process models (SBP models). The goal of the semantic annotation is to explicitly specify the semantics of the tasks and decisions in the process flow. What the tasks are supposed to accomplish, is thus no more specified just in natural language, but explicitly by referencing ontology concepts. The main benefit of the semantic annotation in general is the enablement of automatic semantic-based discovery, which can for example later be exploited to automatically search for Semantic Web Services, which could implement a task in the process, or to find similar process fragments, as described below. The semantic annotation of process models is a prerequisite for all semantic-related functionalities in the following phases of the SBPM lifecycle.

In the following, we describe functionalities or use cases in SBP Modeling, which an SBPMS should support.

- **Semantic annotation of process models:** Same as in conventional BPM, the business analyst uses a well-known flowchart-like notation, such as BPMN, to model processes. While drawing the process elements and specifying the process flow, the business analyst annotates the process elements by referencing ontology entities. Different types of ontologies are relevant to business process management [HR07], e.g.: an organizational ontology is used to specify by which organizational entities tasks are to be performed, a Semantic Web Service (SWS) ontology to specify the IT services that implement tasks, and domain ontologies to describe data used in the processes. By pointing to ontology entities the semantics of the process elements are specified explicitly in a machine-readable manner. The process itself is defined based on a process ontology. The ontologies are created by ontology engineers, domain experts and business analysts. Besides the ontology framework presented in [HR07], there exist also other works in context of enterprise ontologies [Di06, Gr00], which could be used or adapted for SBPM.

In the modeling phase, the semantic annotation of process models enables (or enhances) additional functionalities, namely the discovery and auto-completion of process fragments, which lead to more effective modeling with respect to the reuse of existing process artifacts, as described next.

- **Reuse of process fragments:** Process fragments are parts of a process model which have been identified as potentially reusable. The business analyst can select parts of SBP models and save them as process fragments in a semantic business process repository for later reuse.

  Before or during modeling the business analyst can search for existing process fragments. As a business model may get quite complex, the analyst wants to avoid duplication of work and tries to reuse already existing process fragments. The fragments and models are stored persistently in the process repository and are discovered using semantic-based discovery. The business analyst describes the functionality of the process fragment, which he wants to obtain, by means of a graphical user interface (specifying e.g. the domain of the process, functionalities it contains etc.) and pointing to ontology entities as in the annotation step. After automatic semantic-based discovery, he can then select one alternative and paste it into the process model.
- **Auto-Completion:** During modeling, the analyst can use a special kind of process fragment discovery, the so called auto-completion functionality, well known from the integrated development environments (IDE). The business analyst chooses a part of the process model which is not yet completely modeled. After triggering the auto-completion, the system searches automatically for stored process fragments which could be used to complete the unfinished process.

3.2 Semantic Business Process Implementation

In the previous section, we have described modeling of semantic business processes from the business point of view. In the Semantic Business Process Implementation phase the semantic business process model is transformed to an executable process model, which can be deployed to a process engine for execution.

The transformation of the process description is needed, as the semantic business process model, which was created during the modeling phase, does not contain all necessary information that would allow for its execution. Moreover, the structure of the process may not be well-formed in the sense, that it cannot be represented as a set of instructions to be executed using existing web services. The transformation step involves finding Semantic Web Services, which implement the tasks in the process, specifying data flow, and generating a process model representation that the process engine understands.

The semantic annotation of the SBP model from the modeling phase enables more automation in the implementation phase. Based on the ontological annotation of tasks, corresponding semantic web services can be discovered automatically in an SWS repository. In case no appropriate SWS can be found, the system can use AI planning techniques and try to compose a set of SWS, which satisfy the requirements of the task [We07, WMD+07]. Without semantics, these tasks have to be manually performed by an IT engineer.

The Semantic Business Process Implementation phase requires following additional functionalities:

– **SWS discovery:** An SWS repository stores SWS descriptions and supports semantic-based discovery of SWS. The semantic annotation of a process task is taken as input and compared to the SWS descriptions.
– **Process composition:** Process composition is responsible for the automatic discovery of an SWS or of a composition of several SWSs and process fragments that together implement a task within the process. After a business analyst has finished modeling the process, he requests the system to generate the executable process model. The request is passed to the composition functionality, which uses SWS discovery features to retrieve the relevant SWSs from the SWS repository and/or to find already composed process fragments in the semantic business process repository for each task in the process. If no single SWS can be found, the composition functionality triggers the composition algorithm to derive a SWS composition, which collectively implements the task. Having found an optimal solution, the SBP process model is updated with information on the SWSs or the compositions that implement each task. Furthermore, after checking the correctness of the process it is stored in the semantic business process repository.
– **Manual refinement:** Although the automation of the entire semantic business process implementation is strongly desired, in some cases, the generated process models may need to be refined by IT engineers. They may need to specify some technical aspects like transaction boundaries and security aspects. It may also happen that the discovered services or process fragments might not have the interfaces and data we expect. In that case process and data mediators may need to be created.

- **Process deployment:** After process composition, the SBP process model has to be translated to an executable process model, which can be deployed on a process engine. In addition to the executable process model, an SWS description of the process is generated. The process itself is exposed to the outside as a SWS, and thus its SWS description has to be additionally stored in the SWS repository.

3.3 Semantic Business Process Execution

After the implementation phase, a SBP model is on one hand deployed on a process engine and thereafter it is ready for instantiation and execution. On the other hand, it is externalized as SWS and consequently it is accessible to the clients. The corresponding SWS is an entry point to interact with the SBP and consume its functionalities. The SBP itself uses other SWS to implement its tasks.

Regarding the SBP execution, we can distinguish between three layers similar to the Service Oriented Architecture (SOA for short) [Er05] ones, where the "Service Registry" layer is extended to an infrastructure for SWS execution, the "Service Implementation" layer is more focused on the SBP engine, and the "Service Consumer" layer refers to end user requesting to achieve a goal or to invoke a specific SWS:

- **SBP Engine:** In SOA the "Implementation" layer represents the parties, which implement externalized services and with which a client has to interact in order to consume the requested functionality. In SBPMS this layer is represented by the SBP Engine, which is able to instantiate and execute SBP instances. That does not mean that SBPMS don't consider other kinds of services implementations. However, the SBP Engine should be considered as a first class layer in SBPMS. Services implemented in other way are also considered, however without emphasis on their implementation infrastructures. They are exposed as SWS in the SWS Infrastructure.
- **SWS Infrastructure:** In SOA the "Registry" layer allows hosting services and discovering them according to client criteria. In SBPMS a similar layer is required, however, with more advanced functionalities. Indeed, in order to ensure seamless interaction this layer should provide mechanisms for semantic based discovery, selection and invocation of SWS.
- **Service Requester:** This level corresponds to the end user requesting to achieve a goal or to invoke specific SWS. A SBP engine can play the role of a user requesting to achieve a SBP task.

The main benefit of using SWS in the execution phase is the support of dynamic service binding functionality. The services which are to be invoked out of the process can be determined at runtime by the SWS Infrastructure using semantic-based discovery and then be bound to the process tasks, which they implement. The discovery and selection of the SWS would typically be based on non-functional requirements, such as price or response time. Thus, it is ensured that always the optimal services are invoked. In conventional BPM the used Web services have to be specified at design time, because at runtime it can not automatically be ensured that the discovered Web services, which lack semantic descriptions, have the same functional semantics as the process task, they have to implement. If at runtime the specified Web service is not available or the usage of another Web service would be more appropriate, the process model has to be changed, which is a very time-consuming task.

In the following, we describe the functionalities expected from the SBP Engine and the SWS infrastructure:

- **SBP execution:** The process engine executes a process model by creating a process instance and navigating through the control flow of the process model. A process instance is created when a service requester sends an instantiating message to the process engine, i.e. invokes the process, which is exposed as a SWS. When a task of the process model is to be executed, the process engine delegates the call to the SWS infrastructure.
- **Communication with SWS infrastructure:** The SBP Engine plays the role of a service requester when it invokes the SWS infrastructure in order to perform a SBP task. The SWS infrastructure dynamically discovers an appropriate Semantic Web Service based on the semantic description of the SBP task and invokes it on behalf of the process engine.
- **Achieve Goal:** The “Achieve Goal” functionality is provided by the SWS infrastructure as the entry point for service requesters. It allows to service requesters to send a message to the SWS infrastructure requesting to achieve a specific goal. A goal is a semantic description of the functionality, which is to be achieved. Achieving a goal is subdivided into the following two functionalities:
  - **SWS Discovery and Selection:** In the first step a set of SWS is discovered based on a functional description, and then the best-fitting SWS is selected according to non-functional requirements.
  - **SWS Invocation:** After discovery and selection, the selected SWS is invoked and the invocation result is returned to the service requester. Thereby, the SWS can be implemented as a SBP or as a conventional SWS. The invocation of conventional SWS involves their execution by the backend systems. The execution of SWS implemented by a SBP is performed by the SBP Engine. Technically, from the point of view of the SWS infrastructure, the invocation of the two alternative kinds of SWS implementations does not differ.

3.4 Semantic Business Process Analysis

In Semantic Business Process Analysis we distinguish two different features; the first one is process monitoring which aims at providing relevant information about running process instances in the process execution phase, the second one is process mining that analyzes already executed process instances, in order to detect points of improvement for the process model.

Both process monitoring and process mining operate on the event log which is written by the process engine during process execution. In SBPM, the events communicated are semantically annotated. The semantic annotation is performed on both the level of event payload (e.g. value of a variable) and event type (e.g. defining an event of being an instance of a variable change event). To enable formal classification of events according to event types, an event ontology has to be defined.

Based on the semantic annotation of event payload, reasoning mechanisms can be employed to enable richer monitoring and querying of events.

Process monitoring is the observation and recording of the activities that take place during SBP execution. The monitoring tool gathers information and shows meaningful pieces of it, often in form of dashboards, to the business analyst. There are two kinds of monitoring the SBPMS should support:

- **Passive Monitoring**: Passive Monitoring allows the business analyst to subscribe to events he is interested in; the process engine publishes these events as the process is executed. The business analyst gets the information displayed in a monitoring tool in real-time.
- **Active Monitoring**: Active Monitoring permits the business analyst to actively search for concrete information from the information space. For example, the analyst can search for information in the event logs or he can retrieve further details from the process engine. The business analyst can actively formulate a query in order to retrieve the required information. In SBPM, queries can exploit the semantic annotation of events published in the event logs, and use reasoning mechanisms to deduce implicit knowledge.

Process Mining analyzes business processes based on event logs. The goal of process mining is to help in auditing, analyzing and improving business processes including deriving metrics on the performance of process models such as cost and duration. The event logs contain the complete history of the process instance executions. The events in the event log are ontologically annotated and thus enable reasoning [AA07].

The SBP Mining functionality is provided by one of the following analysis techniques:

- **Semantic Process Discovery**: Process discovery derives the actual executed process model from the event logs. This process model can be compared to the deployed model, showing potentially improvement possibilities.
- **Semantic Conformance Checking**: The defined process model is compared with the process model derived from the event logs. The discrepancies between the log and the model are analyzed. Conformance checking is used to detect deviations, to locate and explain these deviations, and to measure the severity of these deviations.
- **Semantic Organization Mining**: Organization Mining is similar to process discovery, however the focus is on mining of information about social networks in executed processes.
- **Semantic Performance Analysis**: This technique uses the semantic annotations in the process models and in the logs to automatically detect points of improvements, like performance bottlenecks.
- **Semantic Auditing**: This technique allows for checking properties in the event log. This way the analyst can check if the deployed process models meet certain requirements. For doing that, he selects the type of the property he wants to check by defining a new semantic property or by selecting an existing one.

An example on how the semantic annotation of the event logs can be utilized in process mining is shown in [AA07].

4 Conclusion and Outlook

In [HLD+05] the vision of SBPM is presented. The authors state that the degree of automation in bridging the gap between business and IT can be improved by using semantic technologies. As the main issues in bridging between the business and IT perspectives, the authors identify on the one hand the process implementation, i.e. implementing processes which have been specified by business users to run on IT systems, and on the other side querying the process space, i.e. gathering of information on the processes by business users. The vision paper, however, doesn't elaborate in detail on how these issues relate to the current established BPM lifecycle. In this paper, for each phase of the BPM lifecycle, we have identified the required functionalities, which an SBPMS should support, and we have depicted the benefits of using semantics.

In SBPM, process models are semantically annotated during process modeling. In the process modeling phase the semantic annotations enable semantic-based discovery of process fragments and auto-completion of process models. In the process implementation phase process composition functionality exploits semantic descriptions to find SWSs or compositions of SWSs for the implementation of the process. Without semantic descriptions the discovery of appropriate Web services and their composition is a manual task, whereas when using semantics much of the work is automated. During process execution, the use of SWS descriptions in process models enables dynamic binding of services to process tasks. The concrete services, which are invoked by the process, can be selected at runtime, when they are needed, according to criteria such as price or response time. Without SWS, the concrete services have to be specified at design time, which can lead to a non-optimal selection, if alternative better-fitting services are not available until runtime. Finally, in the analysis phase semantically annotated event logs enable reasoning and more powerful querying of events in process monitoring and mining.

In this paper, we have tried to stay technology-independent and to specify requirements, rather than solutions. For example, we have not shown how exactly the semantic annotation of process models should take place, which technologies are used and how. This is part of our ongoing and future work in the context of the SUPER[1] project. There exist already first papers which deal in more detail with composition [WMD+07], process mining [AA07], and relevant ontologies [HR07] in SBPM as developed in SUPER. We are in the process of implementing an SBPMS which will support the functionalities described in this paper. It will be based on, among others, BPMN, BPEL and WSMO technologies.

Acknowledgements

We would like to thank Jesús Contreras, Dimka Karastoyanova, Tammo van Lessen, Frank Leymann, Jörg Nitzsche, Marek Wisniewski and all other colleagues from the

[1] Semantics Utilized for Process Management within and between Enterprises (SUPER) www.ip-super.org

SUPER research project for valuable discussions. This work has in part been funded through the European Union's 6th Framework Programme, within Information Society Technologies (IST) priority under the SUPER project (FP6-026850, http://www.ip-super.org).

References

[AA07] Alves de Medeiros, A.K.; van der Aalst, W.M.P.: Process Mining Towards Semantics. 2007

[ADH+03] van der Aalst, W.M.P; van Dongen, B.F.; Herbst, J.; Maruster, L.; Schimm, G.; Weijters, A.J.M.M: Workflow mining: A survey of issues and approaches. DKE 47 (2003).

[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

[Di06] Dietz, Jan L. G.: Enterprise Ontology. Springer, Berlin / Heidelberg 2006.

[Er05] Erl, Thomas: Service-Oriented Architecture (SOA): Concepts, Technology, and Design. Prentice Hall PTR, 2005.

[Gr00] Gruninger, Michael et al.: Ontologies to Support Process Integration in Enterprise Engineering. In: Computational & Mathematical Organization Theory 6 (2000) 4, pp. 381-394.

[HLD+05] Hepp, Martin; Leymann, Frank; Domingue, John; Wahler, Alexander; Fensel, Dieter: 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

[LR00] Leymann, Frank; Roller, Dieter: Production Workfl–w - Concepts and Techniques. PTR Prentice Hall, 2000.

[RLK+06] Roman, Dumitru; Lausen, Holger; Keller, Uwe; et al.: D2v1.3 Web Service Modeling Ontology (WSMO). WSMO Final Draft 21 October 2006. http://www.wsmo.org/TR/d2/v1.3/, retrieved Apr 09, 2007.

[SF03] Smith, Howard; Fingar, Peter: 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.

[We07] Weber, Ingo: Requirements for the Implementation of Business Process Models through Composition of Semantic Web Services. Proceedings of the 3rd International Conference on Interoperability for Enterprise Software and Applications (I-ESA) March 2007, Funchal, Portugal

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