Martin Hepp, Knut Hinkelmann, Dimitris Karagiannis,
Rüdiger Klein, Nenad Stojanovic (Editors)

Proceedings

of the

Workshop on Semantic Business Process and Product Lifecycle Management (SBPM 2007)

in conjunction with the

3rd European Semantic Web Conference (ESWC 2007)

Innsbruck, Austria, June 7, 2007

Organizing Committee:

Martin Hepp (DERI, University of Innsbruck)
Knut Hinkelmann (University of Applied Sciences Northwestern Switzerland)
Dimitris Karagiannis (University of Vienna)
Rüdiger Klein (Fraunhofer IPK)
Nenad Stojanovic (FZI, University of Karlsruhe)

Workshop URI:
http://sbpm2007.fzi.de/

The proceedings will also be published in the CEUR Workshop Proceedings series, ISSN 1613-0073, available online at

http://CEUR-WS.org/Vol-251/.

Table of Contents

Preface

In the past year, Semantic Web technology has gained a substantial interest from both the Business Process Management (BPM) and the Product Lifecycle Management (PLM) domain. This trend has been further fueled by the growing maturity of ontology languages, tools, and infrastructure, and by the engagement of major software vendors in semantic technology. Many researchers and practitioners are now working on how Business Process Management, in particular in service-oriented environments, may benefit from Semantic Web Services frameworks and other results from Semantic Web research. In general, there is substantiated hope that applying ontologies to core problems of BPM and PLM may bring a breakthrough in terms of agility in changing markets and the management of compliance with customer needs and regulatory requirements.

With this workshop, we want to bring together experts from various communities, namely the Business Process Management community on one hand and the Semantic Web / Semantic Web Services community on the other. In particular, we aim at bundling experiences and prototypes from the successful application of Semantic Web technology to BPM and PLM in various industries, like automotive, engineering, chemical and pharmaceutical products, and services domains.

We received a broad spectrum of submissions from various communities and are confident that the seven full papers and the seven position papers that we eventually selected for publication and presentation will contribute to a better understanding of how the vision of Semantic Business Process Management [1] can be made a reality. All papers were reviewed by at least two members of the Program Committee.

The organizers would like to thank all authors for their submissions and the members of the Program Committee for their time in reviewing the papers.

Martin Hepp
Knut Hinkelmann
Dimitris Karagiannis
Rüdiger Klein
Nenad Stojanovic

[1] Martin Hepp, Frank Leymann, John Domingue, Alexander Wahler, and Dieter Fensel: *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.

Workshop Committees

Organizing Committee and Workshop Co-Chairs

Martin Hepp
Knut Hinkelmann
Dimitris Karagiannis
Rüdiger Klein
Nenad Stojanovic

Program Committee

Andreas Abecker
Roxana Belecheanu
Christian Brelage
Christoph Bussler
Jorge Cardoso
Oscar Corcho
Tommaso Di Noia
John Domingue
douglas foxvog
Fausto Giunchiglia
Nicola Guarino
Michel Klein
Ruediger Klein
Juhnyoung Lee
Jan Mendling
Barry Norton
Markus Nüttgens
York Sure
Oliver Thomas
Yuxiao Zhao

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 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.

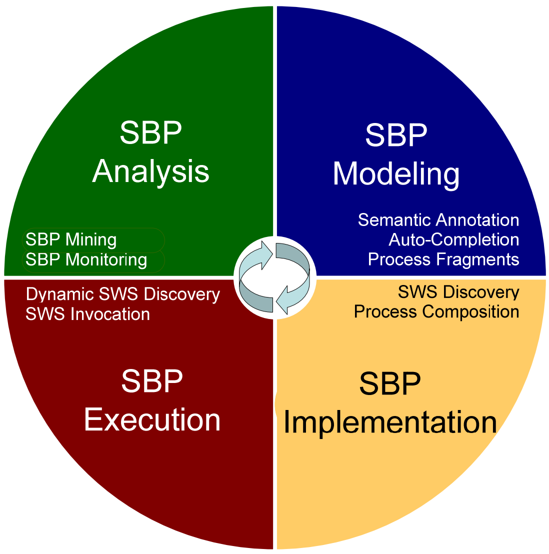

Figure 1: SBPM Lifecycle

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

A Survey on Workflow Annotation & Composition Approaches

Florian Lautenbacher, Bernhard Bauer

Programming Distributed Systems Lab
University of Augsburg, Germany
[lautenbacher|bauer]@informatik.uni-augsburg.de

Abstract: Efficient business processes are key to economic success. With the need to frequently adapt or restructure business processes and workflows in a dynamic market, agile processes and (semi-)automatic workflow and process composition would be useful. Currently, this is a manual and time-consuming task. Automating and optimizing this task is of high interest in research communities. Nevertheless, the orientation differs: Some focus on semantic web services, others on Grid workflows or concentrate on business process management. In this paper we present a survey of available workflow annotation and composition approaches in all of these areas. We additionally categorize and compare them and describe future work.

1 Introduction

In times of dynamic shifting markets, companies, especially those integrated in electronic supply chains, have to adapt or even restructure their business processes frequently. In the last years more and more companies apply the service-oriented approach (SOA) to obtain highly flexible and agile business processes. Using e.g. web services ensures loosely coupled components and hence enables a faster reaction to new business requirements. But still, these changes need to be done manually. However, in the research community there are several attempts to (semi-)automatic workflow composition.

Firstly, the tasks in a workflow have to be annotated with semantic information. Semantic annotation is mostly proposed in literature to annotate documents and web pages. In *Merriam-Webster online* it is defined as “(1) a note added by way of comment or explanation and (2) the act of annotating”. Similarly, Euzenat [1] formalized semantic annotation in the context of the Semantic Web: from two sets of objects, documents and formal representations, two functions can be created: a function from document to formal representations, called annotation and a function from formal representations to documents called index. The corresponding activities are annotation and indexing. Anotation can take place in a descriptive way (plain text) or in a formal way using an underlying logic. The semantic annotation can either be embedded in the workflow itself or can exist as an ontology outside the workflow (e.g. using the TOVE-ontologies [2]).

Automatic workflow composition can enhance workflow reuse and workflow repurposing. According to [3] workflow reuse enables the sharing of workflows by a community as best practices and reduces workflow authoring time, improve the quality and experimental provenance (in e-Science). In workflow repurposing user take workflow fragments that are close enough to be the basis for a new workflow and make small changes to its structure to fit to a new purpose (customizing). Workflow composition can be done with semantic annotated process actions which are annotated with concepts of an ontology and which are then composed to a complete workflow according to a specified goal (Figure 1 shows a small example of a travel expense process).

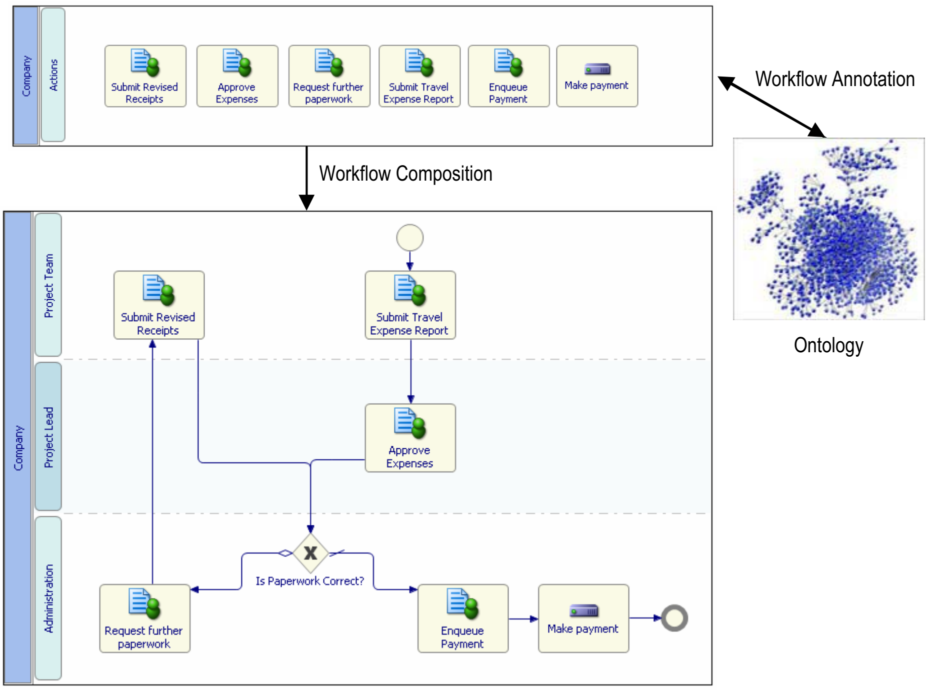

Figure 1: Correlation between workflow annotation and composition

Therefore, to get a deeper understanding of the different research areas as well as on differences and similarities, we describe in this paper the most significant approaches or projects for annotation and composition in the domains business process management, web service and Grid computing and compare them according to predefined requirements. Business process management focuses more on high level processes whereas web services and Grid computing are more concerned with the technical details of processes.

This paper is organized as follows: section 2 gives a short overview about the different domains web services, semantic web, Grid computing and business process management and describes ongoing projects which are going to combine these areas and cover aspects like process annotation or composition. In section 3 we describe the requirements which are used to compare the different approaches. In section 4 and 5 we introduce existing workflow annotation and composition methods. We show a comparison of these approaches in section 6 and describe further research in 7.

2 Workflow in web services, business processes and Grids

Workflow has been defined as “the automation of a business process, in whole or part, during which documents, information or tasks are passed from one participant to another for action, according to a set of procedural rules.” [4]. Business Process Management and workflow are seen as essentially the same, albeit with some differences on emphasis [5]. The same can be said for web services or grid services. A business process describes the actions of an enterprise without any technical information. A workflow might include some technical information, but can still be platform independent. Web services and Grid already describe the platform which is used and all technical information for the invocation. Henceforth, we use the concept ‘workflow’ as collective term for business processes and the orchestration of web and Grid services.

Workflow composition consists of those activities required to combine and link existing workflow fragments and other components to create new processes. This definition is similar for service composition [6] and for Grid workflow composition. However, the automatic composition of application components is challenging, because it is difficult to capture the functionality of components and data types used by the components [7]. That is where the semantic web community comes into play.

Based on the vision described e.g. in [8] and [9] that the usage of ontologies and semantic web standards can extraordinarily improve current business processes, several projects have been launched to combine research areas like business process management and semantic web & web services.

In EU-funded projects like SUPER (http://www.ip-super.org/) and FUSION (http://www.fusionweb.org/fusion) the consortia aim at the development of innovative approaches for business process management using semantic web standards. Methodologies and integration mechanisms for the semantic integration of heterogeneous sets of business applications, platforms and languages should just as well be developed as business process mediation frameworks including semantic business process modelling environments. These projects are in an initial state and there are no deliverables on the topic of workflow annotation or composition available yet, but they show the importance of the usage of semantic information in workflows. SUPER is based on the DIP project, where the interoperability between workflows and ontologies has already been analysed [10].

ASTRO (http://astroproject.org/) supports the composition of distributed business processes for the entire business process lifecycle. The partners create an automated synthesis of composite web services using BPEL4WS in [11]. They import abstract BPEL4WS processes and generate a composite process using the planner MBP. However, their approach is only based on a syntactical level and semantic annotations are not yet considered at the time of this survey.

Many projects in the grid research area focus on business services and workflow aspects. Some of them have developed languages and composition algorithms which will be explained in the next sections in more detail.

3 Requirements for Annotation and Composition approaches

To give a more detailed overview about each described method and language, we categorize them and compare each approach in the following aspects:

- Language: Which underlying (proprietary) language has been used: was an own language developed or were existing languages adapted and extended?
- Application domain: The focus of the approach is specified. Does the language concentrate on web services only, on grid services or is it mainly focused on business processes?
- Semantics: Are semantic annotations possible and does the language directly or indirectly support ontologies? Thereby it does not matter which language the ontologies have (RDF, OWL, WSML, etc.).
- Annot./Comp.: Is the focus of the approach more on annotation, composition, does it consider both or none of them? Some languages are simply designed to annotate existing standards (e.g. WSDL-S) and might additionally be used to automate a composition, but this is not included in the approach itself.
- Hierarchical: details whether a hierarchical decomposition of activities is possible. This hierarchical decomposition includes workflow views, abstraction levels and visibility of processes and activities.
- Research vs. industrial: Has the approach been developed in the research community or is it an industrial standard?

According to [12] there are 5 key workflow aspects which are widely recognised as essential workflow characteristics: functional, behavioural, informational, organisational and operational aspects. These will be used to differentiate the web service, grid service and business process approaches, too.

- Functional: describes whether the functional aspects like inputs, outputs, preconditions and effects of a service / process are included or not. Each service or process can be annotated with functional attributes to describe the functionality, the state of the world before or after execution and the information space before or after its execution.
- Behavioural: describes the *control flow* and shows whether simple or more complex workflow patterns [13] have been considered in the design of the language. These are, e.g., sequence, parallel split, synchronization for simple workflow patterns or arbitrary cycles, discriminator or deferred choice for complex workflow patterns.
- Informational: The informational aspect is defined by the *data* and *data flow* perspective. The three basic modelling elements are parameters, variables and the data flow itself. This includes type definitions and data passing.
- Organisational: characterizes whether the organization structure of a company can be recorded using the language or not and who is responsible for specific tasks in a workflow.

- Operational: The operational aspect is defined by the workflow application perspective. It depicts e.g. whether different invocation methods and styles are offered (like Web Services, Java objects, WSRF, etc.), how the coupling is defined, whether user interaction is required, etc.

4 Workflow Annotation

There are several workflow annotation methods: Semantic-annotated web services are simply called semantic web services; there are several semantic grid workflow language approaches and there already exist first attempts for the annotation of business process models.

4.1 Web Service Annotation / Semantic Web Services

For orchestrating web services the de-facto standard WS-BPEL [14] can be used. Pistore et al. [15] show an approach to annotate the syntactical BPEL-constructs with semantic information. Analogue the underlying web service description language (WSDL) has been enhanced with semantic descriptions in WSDL-S [16] as well as in SAWSDL [17]. OWL-S [18] on the other side stores the semantic information into a new file, but has a clearly defined grounding to the WSDL-file. SWSF [19] extends OWL-S to first-order logic to accomplish more complex statements. One of the most prominent approaches especially in European countries, WSMO [20], has four main elements: ontologies, goals, web services and mediators.

All these approaches are based on overlapping logics: OWL builds on description logics; SWSF extends this to a first-order logic and WSMO in the direction of Logic Programming. A more detailed description about these standards is out of the scope of this paper and can be found in [21].

4.2 Business Process Annotation

Business process models are widely common to capture the workflow of key processes in companies. Therefore, several graphical notations are available: some use the business process modeling notation (BPMN) which is based on the business process definition meta-model (BPDM), Event-driven process chains (EPCs), simple Petri nets or even UML activity diagrams. To execute a business process one can either use languages like XPDL [22] or do it manually. There are currently first efforts to annotate these languages (the graphical as well as the XML-based languages) with semantic information:

In [23] a proposal to annotate EPCs with semantics (sEPC) is presented which includes four instances of ontologies named Business Ontology, Business Process Concepts, sEPC model and the underlying EPC model. We outlined in [24] that activity diagrams can be annotated with inputs, outputs, preconditions and effects (functional semantics) to start an automatic synthesis of business process fragments. In [25] business processes are modeled using Petri-nets and are aligned with domain ontologies using similarity computation and aggregation. [26] developed a multi meta-model process ontology (m3po) to relate choreographies to workflow models. In the context of the SUPER project several ontologies for different languages such as BPMN, BPEL, EPC, etc. are proposed to cover not only behavioural aspects, but also organisational, functional or data perspectives [27].

4.3 Grid Service Annotation

Several EU-funded Grid projects focus the annotation of Grid workflows and developed new languages for semantic workflow management.

The Akogrimo (Advanced knowledge through the Grid in a mobile world) project is realising a reference architecture and framework that allows the creation of mobile dynamic virtual organizations in a grid infrastructure to bring together the market orientation and pervasiveness of mobile communication technology in everyday life. It aims to develop languages for the semantic description of resources and workflows. However, at the current stage it only uses syntactic BPEL4WS-constructs and composes services based on keyword search.

In the context of the NextGRID project a semantic workflow language is developed which supports managing both low level (concrete) and high level (abstract) workflows. The grid workflow enactment can cope with dynamic insertion of arbitrary business processes at run-time. Therefore, a language model, based on an OWL-S extension was defined as well as composition and substitution rules for services and workflows (incl. a formal representation). This language is called OWL-WS (OWL for Workflow and Services) and includes (additionally to OWL-S) concrete services and workflows whereby composite processes are used for modelling workflows that are not only intra- but also inter-service processes [28].

The K-Wf Grid (Knowledge-based Workflow System for Grid Applications) project introduces a Grid workflow description language (GWorkflowDL) based on high-level Petri nets and XML and focuses additionally on Grid workflow orchestration and a semi-automatic mapping of abstract workflows onto concrete Grid services. GWorkflowDL includes properties to point to external semantic descriptions as e.g. in an ontology.

In OntoGrid a framework for annotating, discovering and composing semantic grid services in a (semi)automatic way was developed. This includes a virtual organization ontology, a semantic grid service ontology, a problem-solving method (PSM) description ontology (functional attributes) and knowledge representation and data types ontologies. The semantic grid service ontology consists of a profile (non-functional attributes), a model (relationship with the PSM ontology) and the description of the choreography.

The Taverna workbench, developed in the myGrid project (www.mygrid.org.uk), allows users to construct complex analysis workflows based on the workflow language SCUFL (Simple Conceptual Unified Flow Language) whereby ‘workflow’ is defined here as the specification and execution of ad-hoc in-silico experiments. The offered services are distinguished in domain services which perform scientific functions and services which are created during workflow design and execution. These services can be found using the Feta Semantic Discovery tool which compares input and output parameters and the function of services and assists the workflow design [29].

The SIMDAT project enhances the SCUFL-language to a BPEL4WS-based workflow language named XScufl/Freefluo which describes the control and data flow. A composition is made using abstract services. After the service matchmaking process has finished the concrete OWL-S services need to be discovered [30].

5 Workflow Composition

Similar to workflow annotation, again, there are three different research areas focusing on the topic workflow composition: web service composition, business process composition and grid service composition.

5.1 Web Service Composition

Web Service Composition can be divided in two main parts: static composition and dynamic composition. Static composition includes orchestration (one service orchestrating the others) and choreography (each service describes its interactions). For modeling orchestration and composition different languages have been developed (e.g. WS-BPEL or WS-CDL [31]). While there are first approaches to automatically generate static compositions [32], most of web service composition approaches rely on dynamic composition using semantic annotations.

Trying to fulfil all requirements for an automated service composition (as e.g. described in [33] or [34]), most of the algorithms only create one path to reach the goal – neglecting that there would be other paths interesting for the whole business process, too.

Web service composition can be performed agent-based (as in [35] or [36]), based on interaction protocols [37], symbolic transition systems [38] or some other kind of logic (e.g. temporal action logic [39] or linear logic [40]).

In [41] a heuristic search algorithm for automated Web Service composition is presented. It enhances current heuristic search algorithms and solves shortcomings such as missing parallel and alternative control (XOR) flows, the creation of new variables and support of non-determinism. A service is described as discrete business functionality in a technical way and the enforced hill-climbing algorithm (which is a forward heuristic breadth-first search in state space) is extended. The heuristic function is adapted to calculate the length of the generated composition. Therefore, the used planning graph consists of two kinds of nodes: fact nodes (represent literals from states) and activity nodes (represent service invocations) and can be grouped in layers. The heuristic function counts the number of activity layers which include parallel and alternative service invocations.

The semantic web community has used planning techniques to address the problem of automated composition of semantic web services, e.g. based on OWL-S descriptions of inputs output, preconditions and effects. In [42] SHOP2, a hierarchical task network (HTN) planner, is employed for Web Service composition. The HTN planner creates plans by task decomposition. Given a list of tasks that have to be achieved and a set of Web Services that accomplish these tasks and are described as atomic, simple or composite processes in OWL-S, the planner builds a plan representing an ordered sequence of Web Services that need to be executed. Thus, the main idea of the authors is that task decomposition is very similar to the concept of process decomposition in OWL-S and therefore is suited for automatic Web Service composition. For this purpose OWL-S process descriptions are transformed to a SHOP2 domain. SHOP2 is then used to build a plan, which afterwards is transferred back to an OWL-S description of an executable (composite) process. As an example the authors describe the composition of a Web Service for the planning of a medical investigation out of other Web Services that e.g. make appointments for single medical treatments.

A more detailed description about other approaches and an overview about the mentioned web service composition approaches can for example be found in [43].

5.2 Business Process Composition

Several synthesis (or composition) algorithms for business processes have been proposed using different graphical notations: In [44] business processes are modeled using Petri-nets and are annotated with domain ontologies using similarity computation and aggregation. The similarity can be measured using syntactical, linguistical and structural differences as further outlined in [45]. This method presumes that there is a repository where all business processes have been stored: the synthesis combines (existing) process chains rather than single actions. In [46] cross-organizational business processes are automatically generated using the SAP Enterprise Service Architecture. Therefore, message elements and domain ontologies are aligned, each process is semantically annotated and possible mappings are generated.

5.3 Grid Service / Grid Workflow Composition

Most of the Grid research projects described above offer not only the possibility to annotate Grid workflows, but also to make a (semi-)automatic composition of workflows. E.g. Akogrimo defines languages for the semantic description of resources and workflows in order to define complex Grid services by composing existing Grid services. A workflow manager service translates a business process in an orchestrated composition of simple and complex services whereby “extensions of BPEL4WS seemed to be most promising” [46]. At the current stage no semantic annotation is included and the search for BPEL templates which correspond to the Business Process is just based on simple keyword search. In the future, semantic annotations in Akogrimo will probably be written in semantic languages such as OWL-S [47].

The ODESGS environment of the OntoGrid project facilitates the handling of large numbers of semantic Grid services by means of its (semi-)automatic discovery in the composition of new ones. It uses problem-solving methods (PSM) and ontologies for describing grid services in a formal and explicit way. The PSM description ontology contains a profile, a model and choreography. The ontology for the description of the PSM is based on the Unified Problem-Solving Method Language (UPML) and enables the PSM to automatically compose new grid workflows [48].

A more detailed overview is out of the scope of this paper. For taxonomies of workflow management systems and composition algorithms for Grid computing, please refer to [7].

6 Comparison of existing approaches

Table 1 shows a categorization and comparison of the mentioned approaches in the beginning of 2007. The attempts are categorized in their application domain: Web Service (WS), Grid or Business Process (BP) focus. 'I' notes that it is an industrial standard, 'R' an academic approach (research) and 'RI' that people both from research labs as well as from industry have been involved. The table depicts whether semantic information is directly included ('+'), not covered at all ('-') or whether there is simply a link to existing ontologies ('±'). The table shows whether the approach focuses on annotation only ('A'), on composition only ('C'), covers both ('AC') or none at all ('-'). It describes whether there is an abstraction level ('±'), no abstraction at all ('-'), whether it is not stated in any document we found ('?') or it includes workflow views, abstraction levels and visibility of processes and activities ('+'). The functional aspect indicates whether inputs and outputs are outlined ('±') or preconditions and effects are also included ('+'). In the behavioural column simple control flow ('±') or more advanced control patterns ('+') or no flow ('-') are represented. The informational aspect demonstrates whether data, type and variable definition is possible ('+') or not included in the standard ('-'). Organizational aspects like the hierarchy in a company are either included ('+') or not ('-'). The operational column shows whether at least one invocation style ('±') or different ('+') are offered.

7 Conclusions and further work

As one can see, none of the approaches fulfils all requirements completely and is constructed for annotation and composition similarly.

Especially the organizational perspective is often neglected, but this is an important aspect for queries and reasoning on responsibilities and workload of employees. Most of the approaches come from research organizations or research projects and support functional and behavioural information. The informational and hierarchical perspective are only covered in some approaches.

The most interesting approaches for workflow annotation and composition in comparison to the defined requirements seem to be the ODESGS ontology of the OntoGrid project, the m3po ontology and maybe in the future the ontologies of the projects SUPER or FUSION. Nevertheless, all of them are still work in progress.

All mentioned research areas are probably converging into one single research field: more and more web services will be available on a Grid in the future and using business process modeling one can also model the control flow of web services (or use model transformations between the abstraction levels). There are attempts in each research community to cover the mentioned aspects, but it is still a long way until standardization is going to be finished and one single standard for all aspects and research areas has been defined. In some cases it seems more promising to focus on one problem field or research area, but in many cases an overall language seems most promising. As one can see in all research communities which include workflow aspects such as business process management, the web service and Grid service community, the necessity of annotating services or process actions has been recognised and there are first attempts of (semi-)automatic composition of workflows in every level of detail.

The identified aspects are considered in our current work on a meta-model for semantic business processes and its model-driven integration with semantic web services [49] which will (e.g. in contrast to m3po) directly include the semantic information into the meta-model and will be ideal for workflow composition.

Table 1: Comparison of the workflow annotation and composition approaches

Project/Organisation	**Language**	**Appl. domain**	**Research/Industry**	**Semantics**	**Annot./Comp.**	**Hierarchical**	**Functional**	**Behavioural**	**Informational**	**Organizational**	**Operational**
OASIS	WS-BPEL	WS	I	-	-	±	±	±	+	-	±
W3C	WS-CDL	WS	I	-	-	-	±	±	+	-	±
Uni Trento	„semantic BPEL“	WS	R	±	A	±	±	±	+	-	±
OWL-S Coalition	OWL-S	WS	R	+	A	±	+	+	+	-	±
W3C WG	SAWSDL	WS	R	±	A	-	+	-	+	-	±
WSMO	WSML	WS	R	±	A	-	+	±	+	-	±
DAML.org / SWSI	SWSL	WS	R	+	A	±	+	+	+	-	±
SHOP2	OWL-S	WS	R	+	C	±	+	±	+	-	±
AKOGRIMO	BPEL4WS	Grid	RI	-	C	±	±	+	+	-	±
NextGrid	OWL-WS	WS, Grid	RI	+	AC	±	+	+	+	-	±
K-Wf Grid	GWorkflowDL	Grid	RI	±	A	-	+	±	-	-	±
OntoGrid	ODESGS	Grid	RI	+	AC	?	+	+	+	+	±
myGrid/SIMDAT	XScufl/Freefluo	Grid	RI	±	AC	-	+	±	+	-	±
Uni Innsbruck/Vienna	A-GWL	Grid	R	-	-	±	+	+	+	-	+
WfMC	XPDL	BP	I	-	-	-	+	±	+	+	+
m3pe	m3po	BP	R	+	A	±	+	±	+	+	+
DFKI	sEPC	BP	R	+	AC	-	+	±	-	-	±
Uni Karlsruhe	"semantic Petri-Nets"	BP	R	+	AC	-	+	±	-	-	±
ASTRO	BPEL4WS	BP	R	-	C	±	+	±	+	-	±
SUPER[1]	several ontologies	BP	RI	+	AC	±	±	±	±	±	±
FUSION[1]	FUSION ontologies	BP	RI	+	AC	±	±	±	±	±	±

[1] Since there are currently no results in these projects yet, but only descriptions of the planned achievements, we categorized all workflow aspects with '±'.

Literature

1. Euzenat, J.: *Eight Questions about Semantic Web Annotations*. In: IEEE Intelligent Systems, January / February 2002, pp. 2-9.
2. TOVE Ontology Project, online available at http://www.eil.utoronto.ca/enterprise-modelling/tove/index.html.
3. Goderis, A.; Sattler, U.; Lord, P.; Goble, C. *Seven Bottlenecks to Workflow Reuse and Repurposing*, in: International Semantic Web Conference (ISWC), Galway, Ireland, 2005.
4. Workflow Management Coalition (WfMC). *Terminology Glossary*, 1999.
5. Hollingsworth, D. *The workflow reference model – 10 years on*, WfMC Chair TC, 2004.
6. Dustar, S. *Service Composition*. In: ICSOC 2003, Trento, Italy, 2003.
7. Yu, J.; Buyya, R. *A Taxonomy of Scientific Workflow Systems for Grid Computing*, SIGMOD Record, Vol. 34, No. 3, September 2005
8. Jenz, D.E. *Ontology-based Business Process Management - Strategic White Paper*, Draft, November 2003
9. 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*, In: IEEE ICEBE 2005, October 18-20, Beijing, China, pp. 535-540
10. Norton, B.; Pedrinaci, C.; Lemcke, J.; Kleiner, M.; Henocque, L.; Vulcu, G.: *Ontology for Web Services Choreography and Orchestration.* DIP Deliveryble D3.9
11. Pistore, M.; Traverso, P.; Bertoli, P.; Marconi, A. *Automated Synthesis of Composite BPEL4WS Web Services*, in ICWS 2005, Galway, Ireland, 2005.
12. Jablonski, S.; Bussler, C. *Workflow Management: Modeling concepts, architecture and implementation*. Int. Thomson Computer Press, 1996
13. van der Aalst, W.M.P; ter Hofstede, A.H.M.; Kiepuszewski, B.; Barros, A.P.: *Workflow Patterns*, in: Distributed and Parallel Databases, July 2003.
14. OASIS.org: *Web Services Business Process Execution Language*, Draft, Version 2.0.
15. Pistore, M.; Spalazzi, L.; Traverso, P. *A Minimalist Approach to Semantic Annotations for Web Processes Compositions*. In: ESWC06, Budva, Montenegro, 2006.
16. Akkiraju, R. et al. *Web Service Semantics – WSDL-S*, November 2005, W3C Member Submission.
17. Farrell, J.; Lausen, H. *Semantic Annotations for WSDL*, W3C Candidate Recommendation
18. Martin, D. et al: *OWL-S: Semantic Markup for Web Services*, November 2004, W3C Member Submission
19. Battle, S. et al. *Semantic Web Services Framework (SWSF) Overview*, September 2005, W3C Member Submission.
20. Lausen, H.; Polleres, A.; Roman, D. (Eds.) *Web Service Modeling Ontology (WSMO)*, June 2005, W3C Member Submission.
21. Aggarwal, R. *Semantic Web Services Languages and Technologies: Comparison and Discussion*, 2004
22. Workflow Management Coalition: *XML Process Definition Language*, WFMC-TC-1025, Version 2.00 Final, October 2005.
23. Thomas, O.; Fellmann, M.: *Semantic Event-Driven Process Chains*. In: Workshop SBPM at the ESWC 2006, Budva, Montenegro, June 2006.
24. Lautenbacher, F.; Bauer, B. *Semantic Reference and Business Process Modeling enables Automatic Synthesis*. In: Workshop SBPM at the ESWC 2006, Budva, June 2006.
25. Brockmans, S.; Ehrig, M.; Koschmider, A.; Oberweis, A.; Studer, R. *Semantic Alignment of Business Processes*, in ICEIS 2006, INSTICC Press, Paphos, Cyprus, May 2006.
26. Haller, A.; Oren, E.; Kotinurmi, P. *m3po: An Ontology to Relate Choreographies to Workflow Models*, Service Computing Conference (SCC), 2006.
27. Hepp, M.; Roman, D. *An Ontology Framework for Semantic Business Process Management*, Proceedings of Wirtschaftsinformatik 2007, Karlsruhe, March 2007.

28. Beco, S.; Cantalupo, B.; Matskanis, N.; Surridge, M. *Putting Semantics in Grid Workflow Management: the OWL-WS approach*, SemanticGrid.org, January 2006.
29. Lord, P.; Alper, P.; Wroe, C.; Goble, C. *Feta: A Light-Weight Architecture for User Oriented Semantic Service Discovery*, In: ESWC, Heraklion, Greece, 2005.
30. Qu, C.; Zimmermann, F. *Application of Standard Semantic Web Services and Workflow Technologies in the SIMDAT Pharma Grid*, W3C Workshop on Frameworks for Semantics in Web Services, Innsbruck, Austria, June 2005.
31. Kavantzas, N.; Burdett, D.; Ritzinger, G.; Fletcher, T.; Lafon, Y. *Web Services Choreography Description Language Version 1.0*, W3C Candidate Recommendation, November 2005.
32. Meyer, H.; Kuropka, D. *Requirements for Automated Service Composition*, in: *Business Process Management Workshops*, LNCS 4104, p. 439-450.
33. Missikoff, M.; Schiappelli, F.; Taglino; F. *A Controlled Language for Semantic Annotation and Interoperability in e-Business Applications*. Semantic Integration (SI-2003) at the ISWC 2003, Florida, USA, October 2003.
34. ter Beek, M.; Bucchiarone, A.; Gnesi, S. *A Survey on Service Composition Approaches: From Industrial Standards to Formal Methods*; Technical report, 2006-15.
35. Küngas, P. *Distributed Agent-Based Web Service Selection, Composition and Analysis through Partial Deduction*, Ph.D. Thesis.
36. Ermolayev, V.; Keberle, N.; Kononenko, O.; Terziyan, V. *Proactively Composing Web Services as Tasks by Semantic Web Agents.* Chapter in JWSR.
37. Wu, Z.; Harney, J.F.; Verma, K.; Miller, J.A.; Sheth, A.P. *Composing Semantic Web Services with Interaction Protocols*, Technical report.
38. Pathak, J.; Basu, S.; Honavar, V. *Modeling Web Service Composition using Symbolic Transition Systems*, in: AAAI Workshop AI-SOC 2006.
39. Giordano, L.; Martelli, A. *Web Service Composition in a Temporal Action Logic*, AI for Service Composition (AISC2006).
40. Rao, J.; Küngas, P.; Matskin, M. *Composition of Semantic Web services using Linear Logic theorem proving*, Elsevier Information Systems 31 (2006) 340 – 360
41. Meyer, H.; Weske, M. *Automated Service Composition using Heuristic Search*, in BPM 2006, Vienna, Austria, LNCS 4102, p.81-96
42. McIlraith, S.; Son, S. *Adapting Golog for Composition of Semantic Web Services,* in: Knowledge Representation'02, 2002.
43. Berardi, D.; De Giacomo, G.; Mecella, M.; Calvanese, D. *Automatic Web Service Composition: Service-Tailored vs. Client-Tailored Approaches.* In: AISC, Beijing, 2006.
44. Brockmans, S.; Ehrig, M.; Koschmider, A.; Oberweis, A.; Studer, R. *Semantic Alignment of Business Processes*, ICEIS, Paphos, Cyprus, May 2006.
45. Betz, S.; Kling, S.; Koschmider, A.; Oberweis, A. *Automatic User Support for Business Process Modeling.* In: Workshop SBPM at the ESWC 2006, Budva, June 2006.
46. Laria, G. *Akogrimo D4.4.2: Prototype Implementation of the Grid Application Support Service Layer*, 30.01.2006..
47. Wesner, S.; Jähnert, J.M.; Escudero, M.A.T. *Mobile Collaborative Business Grids – A short overview of the Akogrimo Project.* Akogrimo White Paper
48. Goble, C.; Gomez-Perez, A.; Gonzalez-Cabero, R.; Perez-Hernandez, M.S. *ODESGS Framework, Knowledge-based Annotation and Design of Grid Services*, ICSOC 2005, Amsterdam, The Netherlands.
49. Lautenbacher, F.; Bauer, B. *Creating a Meta-Model for Semantic Web Service Standards,* In: Proceedings of WEBIST 2007, Barcelona, March 2007.

Putting Business Intelligence Into Documents

Tobias Bürger[1,2]

[1] Salzburg Research, Jakob-Haringer-Str. 5/III, 5020 Salzburg, Austria,
tobias.buerger@salzburgresearch.at
[2] Digital Enterprise Research Institute (DERI), Technikerstrasse 21a, 6020 Innsbruck, Austria,
tobias.buerger@deri.org

Abstract. Business processes are often statically implemented and may not be established ad-hoc. For the realization of dynamic process configurations that demand for changes in these implementations static implementations are not suitable. In this paper we present our ideas on enabling dynamic business process implementations by reverting competencies in today's business processes, i.e. away from the system to the document that is processed. Our idea is to add semantics to business processes by modeling them as a facet of so-called Intelligent Content Objects. We present our ideas of mapping these task descriptions to current business process standards and the Web Service Modeling Ontology (WSMO) to make it useful in workflow execution environments like BPEL4WS and in Semantically Enhanced Service Oriented Architectures based on Semantic Web Services (SWS).

1 Introduction

The intention of Business Process Management (BPM) is to manage the execution of business processes based on a business expert's view. Several drawbacks exist for mediating between these experts' views and the resulting implementations which could be resolved by applying semantics to BPM which is already shown in [5]. In [5] is shown that besides other reasons the lack of machine-readable representations is a major obstacle towards mechanization of BPM. An additional technical obstacle for porting document centric processes between different systems could be overcome by bundling process descriptions with documents which are going to be processed, ie. to apply Intelligent Content Objects which include formal descriptions of their included content and declarative process descriptions. These descriptions may capture an ontological representation of the expert's view which enables its conversion to BEPL4WS and the Web Service Modeling Ontology (WSMO). This conversion would make our approach compatible with the proposal of Semantic Business Process Management (SBPM) by Hepp et al. [5].

2 The Role of Intelligent Content Objects in SBPM

2.1 Intelligent Content

The term Intelligent Content (IC) is a notation for content containing information with explicit semantic descriptions of its properties. Intelligent Content Models as previously assessed for example in [2], can be seen as a carrier for semantically rich information goods which include all the information that is needed to deal with the content in specific situations: Imagine for example a scenario where a contract goes through several validations, additions, modifications and other operations from various people in the course of a workflow. This contract, included content, and the description of the workflow we see as parts of an IC Object. Having the contract together with descriptions of its associated processes is useful when the content is processed in foreign systems which are not knowing how to deal with the content or how to query its properties beforehand. To apply IC Objects in SBPM frameworks is especially useful for content procurement and billing processes. What is essential for making IC Models useful in BPM however, are task and process descriptions available in a declarative and formal form. We intend to follow the KCO approach [1] which includes such descriptions based on the DOLCE Plans and Tasks Ontology (DDPO)[3] [4].

2.2 KCO – A Model for Intelligent Content

Knowledge Content Objects (KCOs) are based on the DOLCE foundational ontology[4] and have so-called semantic facets that form modular entities to describe the properties of KCOs, including the raw content object, metadata and knowledge specific to the content object and about the subject matter of the content. In addition to this knowledge structure the KCO defines a structure based on the different domains of the knowledge objects. This structure is divided into six so-called *facets*, each of them optimized for a specific usage (see [1] for details):

1. **Content Description** includes access information, meta data schemes and subject matter knowledge
2. **Presentation Description** describes how the content (and the knowledge) of the KCO is presented to users and specifies modes of interaction
3. **Community Description** contains descriptions of plans, tasks, roles and goals in the context of a community , and a list of actions performed during the content lifecycle.
4. **Business Description** specifies how to trade the content, including the specification of business models and negotiation protocols.
5. **Trust and Security** specifies methods that ensure security and trust for KCO users

[3] DDPO is an extension of DOLCE (Descriptive Ontology for Linguistic and Cognitive Engineering), DnS (Ontology of Descriptions and Situations), and Plans Ontologies.
[4] http://www.loa-cnr.it/DOLCE.html

6. **Self-description** declares the structure of the KCO itself, including active facets, ontologies used, etc.

The use of foundational ontologies in KCOs establishes a minimal but shareable model for content interoperability between heterogeneous applications. In particular they are the basis for a common understanding of the structure of information, enable the reuse of domain knowledge, make assumptions explicit and enable to separate domain knowledge from operational knowledge.

2.3 Application of KCOs in SBPM

The main intention of SBPM is to increase the level of automation in BPM by representing the various spheres of an enterprise using ontology languages and Semantic Web Service Frameworks [5]. The authors of [5] aim to (1) semantically represent and describe processes, (2) to ontologically capture the IT landscape and domain knowledge, (3) to create a semantic integration layer for transactional data, (4) to perform semantic search on processes, data, and resources, and (4) to use SWS execution environments for the mediation between business goals and business expert's needs. The intention of KCOs – as introduced in section 2.2 – is also to model parts of the process space of BPM using foundational ontologies : (1) User Tasks in the context of a community that use KCOs (in its community facet), (2) User roles in the particular community (in its community facet), and (3) negotiation protocols and pricing schemes for content negotiation (in its business description facet).
To apply IC Objects in SBPM systems like the one proposed in [5], a mapping from DPPO to WSMO is needed in order to enable the execution of plans modeled in the business facet of a KCO. This mapping is conceptually possibly as it was shown indirectly in [7] where an alignment of OWL-S to DOLCE is reported and by Scicluna et. al in [8] who map OWL-S to WSMO. We intend to combine these two approaches to implement a declarative mapping layer which then shall enable the execution of business processes associated with a KCO.

2.4 Expected Benefits

KCOs are a tool to capture and model essential knowledge about a particular entitiy or situation (the "knowledge") in one place and they can be used to transfer this knowledge and content between heterogeneous systems. But introducing KCOs in SBPM frameworks is not only useful when transferring them between different applications: They can also be seen as a communication tool, ie. they may capture a problem in a domain and the possible solutions and processes available for that problem. This information can be visualized (using information in the presentation facet) to increase knowledge transfer.

2.5 Related Work

The TOVE project has developed a set of ontologies for describing various aspects of an enterprise [3] and Hepp et. al introduce an ontology infrastructure for SBPM in [6]. The ontologies used in the KCO are related to the ones in [3, 6], as they are used to model basic notions of processes, activities, particulars and their roles (which are part of the Upper Process Ontology, Upper Organizational Ontology, Business Functions Upper Ontology in [6]). There a three levels of KCOs: generic, domain, and application level KCOs. The foundational ontologies used for (a generic-level) KCOs are intended to be refined to develop specific domain and application ontologies like the Business Organization Ontology in [6] for domain- and application-level KCOs.

3 Conclusions and Future Work

In this position paper we provided arguments for the usefulness of Intelligent Content Objects in Semantic Business Process Management (SBPM). They provide a minimal shareable model for content interoperability between heterogeneous systems and enhance the transfer of business knowledge and content between them. Future work includes the realization of the mapping between DDPO and WSMO in the project GRISINO[5] whose goal is to demonstrate the usefulness of the combination of Intelligent Content and Semantic Web Services.

References

1. Wernher Behrendt, Aldo Gangemi, Wolfgang Maass, and Rupert Westenthaler. Towards an ontology-based distributed architecture for paid content. In A. Gomez-Perez and J. Euzenat, editors, *Proceedings of ESWC 2005*, 2005.
2. Tobias Buerger, Ioan Toma, Omair Shafiq, and Daniel Doegl. State of the art in sws, grid computing and intelligent content objects - can they meet? Technical report, GRISINO Deliverable D1.1 (public), 2006.
3. M.S Fox. The tove project: A common-sense model of the enterprise. In F. Belli and F.J. Radermacher, editors, *Industrial and Engineering Applications of Artificial Intelligence and Expert System*, number 604 in LNAI, pages 25–24. Springer, 1992.
4. Aldo Gangemi and Peter Mika. Understanding the semantic web through descriptions and situations. In *Proc. of the International Conference on Ontologies, Databases and Applications of Semantics (ODBASE 2003)*, 2003.
5. Martin Hepp, Frank Leymann, Chris Bussler, John Domingue, Alexander Wahler, and Dieter Fensel. Semantic business process management: Using semantic web services for business process management. In *Proc. of the IEEE ICEBE*, 2005.
6. Martin Hepp and Dimitriu Roman. An ontology framework for semantic business process management. In *Proceedings of the 8th international conference Wirtschaftsinformatik 2007*, pages 423–440, 2007.
7. Peter Mika, Marta Sabou, Aldo Gangemi, and Daniel Oberle. Foundations for owl-s: Aligning owl-s to dolce. In *Papers from 2004 AAAI Spring Symposium - Semantic Web Service*, pages 52–60, 2004.
8. J. Scicluna, R. Lara, A. Polleres, and H. Lausen. Formal mapping and tool to owl-s. Wsmo working draft, DERI, 2004.

[5] http://www.grisino.at

Ontology-based representation of compliance requirements for service processes

Rainer Schmidt
Aalen University
Beethovenstraße 1, 73430 Aalen, Germany
Rainer.Schmidt@htw-aalen.de

Christian Bartsch
Information Process Engineering (IPE)
Research Center for Information Technologies (FZI)
Haid-und-Neu-Str. 10-14, 76131 Karlsruhe, Germany
bartsch@fzi.de

Roy Oberhauser
Aalen University
Beethovenstraße 1, 73430 Aalen, Germany
Roy.Oberhauser@htw-aalen.de

Abstract: Service processes are becoming increasingly essential in modern economies as traditional, production-oriented industries decline. When comparing service processes to standard business processes, a major distinction is that the quality of their result, i.e., the service produced, cannot be measured in advance. Therefore, the compliance of the service process with quality standards plays an important role in convincing the customer that the services rendered will result in the quality specified. However, the check for compliance is still a tedious task. To address this situation, an ontology-based approach for representing service processes and checking their compliance is proposed. It is based on two ontologies: one to represent the service processes and the other to store the compliance requirements. The process representation ontology uses three so-called views to appropriately represent the service processes. The ontology for storing the compliance requirements differentiates syntactic, semantic and pragmatic requirements.

1 Introduction

Service processes are processes that produce services. Their significance is increasingly growing in modern economies as traditional, production-oriented industries decline. An important difference of service processes when compared to standard business processes is that the quality of their outcome, i.e., the service produced, cannot be measured beforehand [BuSc06]. This is due to the fact that services cannot be produced in advance because they cannot be stored. On the contrary, material products can be tested before they are used since it is possible to store them. Therefore, it is vital to convince the potential customer that the services rendered will result in the quality specified [BuSc06]. The quality of a service to be provided can only be estimated by checking if the process used to provide it is in compliance with quality standards, such as ISO 20000 [ISO20000]. Standards such as ISO 20000 are a kind of abstract specification which define compulsory elements and structures of a process necessary to provide the services in the quality required. However, they do not specify the service processes in full detail because there are different ways to achieve a certain level of service quality. Thus, there is a large divide between the quality standard and the service process, similar to the IT / process divide described in [HLDW05]. As a consequence, the effort to check whether a service process is in compliance with standards such as ISO 20000 is rather high.

Thus this paper provides an approach to appropriately represent compliance requirements and to provide a vision how to reduce the effort for checking the compliance of service processes with quality standards such as ISO 20000. Two ontologies [Grub95] are used. The process ontology is the basis for the ontology-based representation of the service process; the compliance ontology represents the compliance requirements the service process has to fulfill. The paper proceeds as follows: Section 2 shows how to represent service processes using an ontology-based approach. Section 3 defines the representation of compliance requirements in an ontology. The checking for compliance and the implementation is described in Section 4. Section 5 deals with the analysis of related work. Finally a summary and an outlook are given.

2 Process Ontology

As already stated in [HeRo07], a process is more than the mere connection of activities. Thus, it is necessary to reason about the appropriate representation of service processes, because they contain additional elements when compared with standard business processes. Two kinds of additional elements will be identified later on: interactions and resources.

In most business processes the customer is only interested in the outcome of the process but not in the process itself. On the contrary, in service processes, there are many interactions between the service provider and the customer as well as third party service providers. These interactions often have to follow predefined patterns and have to be documented to serve as proof in latter disputes. Thus, it is necessary to appropriately represent these interactions as shown in Figure 1. Interactions connect two activities: one is executed by the service provider and the other by the service client. Between both, a multitude of communication acts may take place which often cannot be fully specified. For example, the clearing of documents requires frequent interactions between the participants. Neither the type of dialog (telephone, e-mail, etc.) nor the interaction frequency can be specified in advance. However, key events, such as the transmission of documents and their clearance can be specified. Therefore, the internal behavior of interaction is defined by event-condition-action rules. They describe the condition to be tested when a certain event occurs and the action to be taken if the condition is met. The starting and ending points of interactions are defined using pre- and post-conditions. The pre-conditions specify the circumstances that need to be fulfilled to start the interaction.

The post-condition specify the results that have to be obtained before the interaction can be regarded as completed.

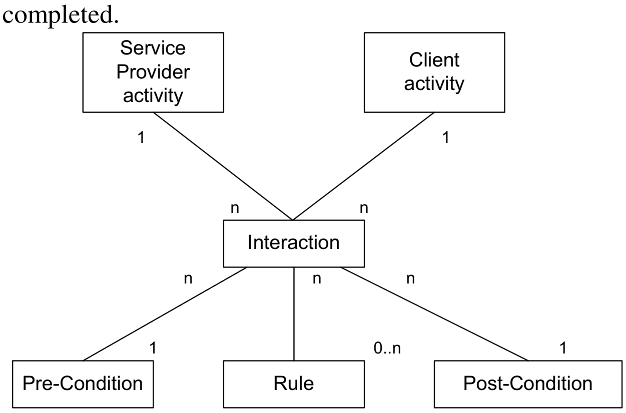

Figure 1: Interactions

Service processes differ from traditional business processes because they extensively use external resources, both from the customer and third party service providers, which have to be appropriately integrated and administered [ZdHe05]. For example, before configuring a customer's computer system, the administrative privileges must have been granted. In addition, if external resources are no longer available but needed for service providing, a procedure to correct these errors has to be started. Finally, customer resources that were used for service providing need to be returned when the service has been completed. In order to properly represent changes in the resource view, adding, changing, and removing resources needs to be a simple process.

The discussion about the proper representation of service processes does not only include the "what" but also the "how", because intermixing independently evolving process elements causes a multitude of side effects. One example for the mixture of views is the "flow dependence" of application programs described by [LeRo97]. Flow dependence means that application programs contain a predefined control flow, making them inflexible to business process changes. To avoid the intermixing of independently evolving functionality, a view-oriented approach is chosen to represent the service processes in an ontology. Views are sets of process elements to mirror aspects of reality evolving completely independent of each other. To appropriately represent service processes, interactions and resources are represented as separate views. Additionally, five basic views are used to organize the process elements common to business processes. They are the functional, operational, control, informational, and organizational view as identified in [Jabl94]. The functional view describes the goal of the process. The operational view specifies activities executed during the process. The control view defines the preconditions for certain activities. The informational view specifies the information exchange that takes place between activities. Schema, schema elements, and relations are the elements of the informational view. The organizational view associates roles with activities. As these views have been extensively analyzed in research, e.g., [JaBS97], [Jabl94], [BKKR03], they are not discussed here.

An ontology for ontology-based process modeling has to reflect the semantics of the service processes. As a result, the process ontology is organized in three layers as shown in Figure 2, displaying a part of the incident management process of ITIL [ITSM04].

The visual representation follows the suggestions made in [BVEL04]. The layers define the abstraction levels of the ontology. The first layer is called meta layer. It defines the concepts for describing the views in the second layer. Using this approach, extensibility is achieved, allowing easy integration of additional views in the second layer via the concepts of the basic layer. The second layer is called view layer. In the second layer, views and their elements are defined using the concepts defined in the first layer. Seven views are defined in the view layer: functional, operational, control, informational, organizational, interaction, and resource. The concrete process element classes are defined in the third layer, which is called the process layer. The instances of the process element classes are used to create the ontology-based process representation.

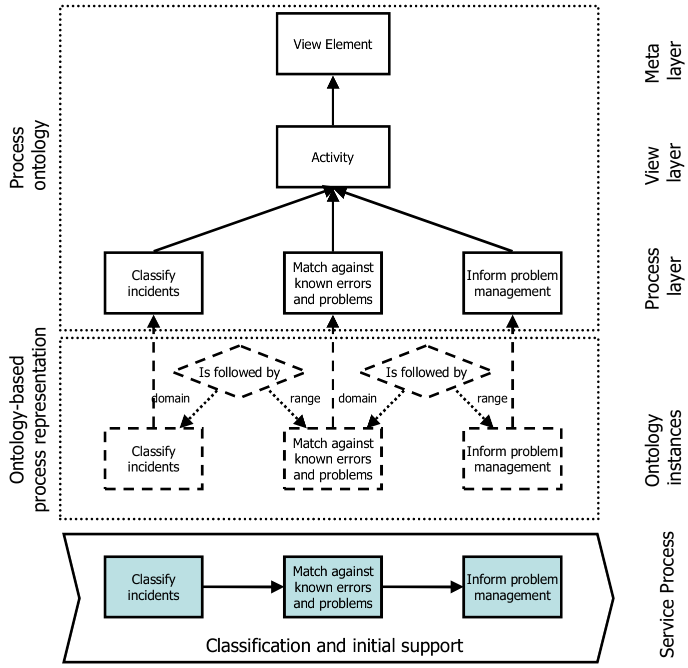

Figure 2: Structure of the ontology

Ontologies are defined using OWL [McHa04]. A one-to-many relationship is used between process element classes in the process layer and the instances in the ontology-based process representation. This is crucial for detecting synonyms and homonyms if a process element appears several times, as shown in Figure 3. The one-to-many relationship permits expressing the semantic identity or non-identity of process elements[1]. Synonyms in the ontology-based process representation can be detected because process elements with different names point to the same concrete process element class. Homonyms can be found by detecting that two process elements of the same name point to different process element classes.

[1] Other approaches such as [ThFe06] use a one-to-one relationship which does not support the detection of synonyms and homonyms.

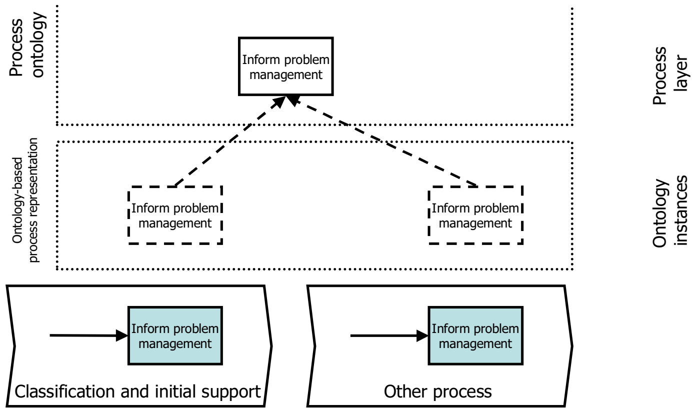

Figure 3: Detection of synonyms and homonyms

The process layer contains not only is-a relationships but also a part-of hierarchy. Is-a relationships allow unambiguously classifying each element of the process model. Part-of relationships allow representing aggregations. For example, it is possible to define the role "Incident Manager" as part of the section "Incident Team".

3 Compliance ontology

Compliance standards, such as ISO 20000, define objectives that have to be fulfilled by the service processes to assure a certain quality of the services provided. These objectives have a set of supporting requirements, which may be syntactic, semantic, or pragmatic. Syntactic requirements contain rules for the description of the service processes. For example, they postulate that each activity needs to be associated with a role, which is responsible for the activity. Semantic requirements demand the existence of certain objects or structures in the process definition. For example, they postulate the existence of a certain kind of document, such as a service report. Pragmatic requirements describe goals that have to be realized within the process. Syntactic and semantic requirements can be verified automatically or at least semi-automatically. Pragmatic requirements can only be verified manually.

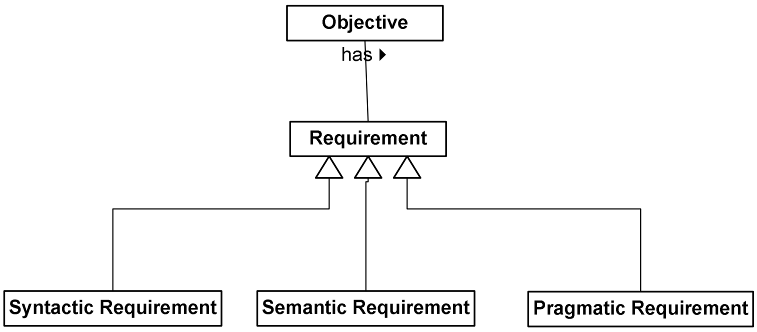

Figure 4: Types of compliance requirements

3.1 Syntactic Requirements

Syntactic requirements are rules defining which model elements have to be used or may be used for the definition of service processes. Therefore, they can be thought of as a "meta model" for the models of service processes. Syntactic requirements can be easily expressed via logic operators. For example, the rule "responsible role required for each activity" can be expressed as follows

$$\text{activity(a)} \Rightarrow \exists\ \text{r}\quad \text{role(r)} \wedge \text{responsible(a,r)}$$

Syntactic requirements can be directly represented in an ontology. E.g., in order to express that a responsible role needs to be assigned to each activity, a property "responsible" is defined, with activity elements as the domain and roles as the range.

3.2 Semantic requirements

Semantic requirements postulate the existence of certain structures or objects in the service process. Semantic requirements can be further differentiated into assertion requirements, structure requirements and action requirements as shown in Figure 5. Assertion requirements define conditions, which have to be matched. It is not specified who has to assure that the condition is met. For example, ISO 20000 defines that "service levels shall be monitored". This can be expressed formally as follows

$$\text{servicelevel(s)} \Rightarrow \text{monitored(s)}$$

Action requirements define actions to be made. An action is comprised of a role, responsible for the action, a verb that describes the action and the object of the verb. For example, ISO requires that "management shall conduct reviews". The responsible role is "management", "conduct" is the verb and "reviews" are the object.
Structure requirements define structures, which have to exist in the process to be compliant with the objective. Structure requirements consist of structure elements, which may be nested. Structure requirements define structures that must exist in the process to

be compliant with the objective. Structure requirements consist of structure elements that may be nested.
For example, ISO 20000 requires the creation of so-called service reports and defines the content of a service report: it should contain the performance towards service level targets, trend information, etc.

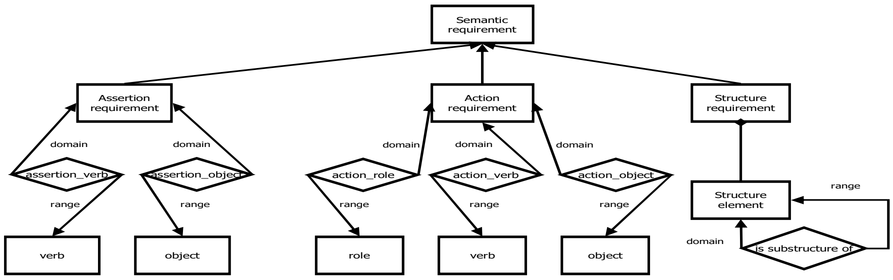

Figure 5: Semantic requirements

The OWL-code to represent the ISO 20000 requirement that the management shall conduct reviews is shown below:

```
<owl:Class rdf:ID="Requirement_Conduct_Reviews">
  <owl:equivalentClass>
    <owl:Class>
      <owl:intersectionOf rdf:parseType="Collection">
        <owl:Restriction>
          <owl:hasValue>
            <Object rdf:ID="Reviews"/>
          </owl:hasValue>
          <owl:onProperty>
            <owl:FunctionalProperty rdf:about="#action_object"/>
          </owl:onProperty>
        </owl:Restriction>
        <owl:Restriction>
          <owl:onProperty>
            <owl:ObjectProperty rdf:about="#action_role"/>
          </owl:onProperty>
          <owl:hasValue>
            <Role rdf:ID="Management">
              <is_member_of>
                <Section rdf:ID="Section_19"/>
              </is_member_of>
            </Role>
          </owl:hasValue>
        </owl:Restriction>
        <owl:Restriction>
          <owl:onProperty>
```

```
          <owl:ObjectProperty rdf:about="#action_verb"/>
        </owl:onProperty>
        <owl:hasValue>
          <Verb rdf:ID="conduct"/>
        </owl:hasValue>
      </owl:Restriction>
    </owl:intersectionOf>
  </owl:Class>
</owl:equivalentClass>
<rdfs:subClassOf rdf:resource="#Action_Requirement"/>
</owl:Class>
```

3.3 Pragmatic requirements

Pragmatic requirements define abstract goals to be achieved. An example is the requirement in ISO 20000 to “communicate the importance of meeting service management objectives and the need for continuous improvement”. The representation of such requirements is very difficult in a machine-readable and especially in a verifiable way. This is because the action outcome does not result in structures that can be represented by a machine, but rather are effects outside the computer system. A possible solution is endorsement. Thus, the effects of actions are not directly measured, but rather the endorsements of people involved.

4 Application scenario

A possible application scenario is the check whether a service process, such as the incident or problem management process [ITSM04], complies with the ISO20000 standard. To do so, the service process is transformed into an ontology-based process representation using the process ontology. For performing the check of compliance, the property of ontologies is used that the membership to an ontology class is not only defined by instantiation, but also declaratively by checking the necessary and required conditions of a class. That means the membership to a class is assigned to an object if the object fulfills all necessary and required conditions of the class. Therefore, all process elements are associated to the compliance ontology, and thus it is checked whether they fulfill the necessary and required conditions of classes in the compliance ontology, as shown in Figure 6. This concept is called realization [SPGK06]. Generally speaking, it is the procedure to find the most specific class an instance belongs to. Realization is supported by reasoners such as Pellet [SPGK06].

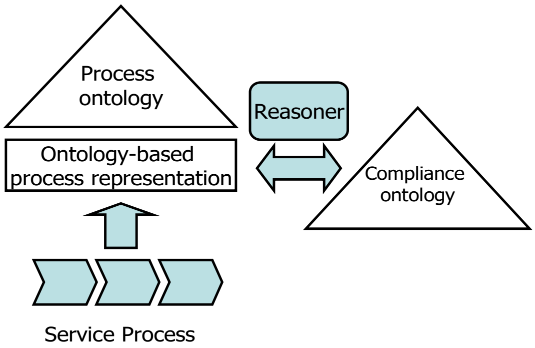

Figure 6: Application scenario

After the reasoner has classified all process elements, the compliance requirements, which are fulfilled, are associated with instances. The compliance requirements, which are not met, have no instances associated. By this means it is possible to determine the set of fulfilled compliance requirements. The concepts defined so far were implemented with the Protégé tool [PROT] and the Pellet reasoner.

5 Related work

There are several areas of related research: the ontology-based representation of business processes and meta-models, the modeling support for service processes, and the view-oriented modeling of processes and workflows.

An important area of related work is the ontology-based representation of business processes and meta-models. An ontology-based approach for formalizing Petri-net-based business processes is given in [KoOb05]. The ontology-based representation of event-driven process chains (epcs) is proposed in [ThFe06]. Both approaches use ontologies for the representation of business processes, but lack the support of service processes.

A first approach to verify business processes against compliance standards has been made in [SoHa06]. A reference model based approach is presented in [KiFo02]. Yet no further details are given on the ontology structure and the procedures for checking compliance. The semantic alignment of business processes using ontologies is described in [BEKO06]. Ontologies are also used to represent the types of modeling methods in [RoGr02], [RoIG04]. Ín [HeRo07] and [HLDW05] and the basic structure for representing business processes by ontologies is described, however no further details are given.

Service process models have been analyzed by two groups of authors. The support and the modeling of service processes using a coarse-grained architecture is discussed in [KlWe01], [WeKl04], [WeKl03]. A view-oriented approach for the modeling of service processes is presented in [Schm06]. In [BöJK03] a modularization approach for services in the information technology business is proposed.

This work is based on modeling business process aspects and its relations to each other utilizing the Unified Modeling Language to allow a simpler operationalization of

business process reference models. [BWFK04] has designed and developed a simple layer-based model for managing service data. The proposed model is a first step towards ontology.
The view concept is also included in modeling methods, such as ARIS [Sche91], or programming concepts, such as aspect-oriented programming [Kicz96]. It has been described for a range of applications, such as for specifying inter-organizational workflows [BKKR03] or supporting business processes and cross-organizational business processes [Schm03].

6 Conclusions

This paper has introduced an ontology-based approach to represent service processes and their compliance requirements. Thus, it lays the foundation for verifying the compliance of service processes. Two ontologies were defined: The process ontology defines the concepts needed to represent service processes. The compliance ontology contains concepts to represent objectives and requirements for compliance standards. Three types of compliance requirements have been identified: syntactic, semantic, and pragmatic. Syntactic requirements can easily be represented by constraining the properties used for connecting the process elements. Semantic requirements can be further differentiated into assertion, action and structure requirements. Assertion requirements define conditions which have to be met. They consist of a verb and an object. They do not specify directly who is responsible for the condition Action requirements define actions to be performed as part of the process. An action is comprised of a verb and an object of the verb. Structure requirements define structures that must exist in the process in order to be compliant. Requirements can be checked by applying a reasoner to the ontology-based process representation. Requirements are checked by the classification of process elements. Compliance requirements, which are instantiated, are met. By this means it is possible to determine the set of fulfilled compliance requirements.

7 References

[BEKO06] Brockmans, S.; Ehrig, M.; Koschmider, K.; Oberweis, A.; Studer, R.: Semantic Alignment of Business Processes. In: Proceedings of the Eighth International Conference on Enterprise Information Systems (ICEIS 2006), pp. 191-196. INSTICC Press, Cyprus, 2006.

[BKKR03] Bernauer, M.; Kappel, G.; Kramler, G.; Retschitzegger, W.: Specification of interorganizational workflows - A comparison of approaches. In: Proceedings of the 7th World Multiconference on Systemics, Cybernetics and Informatics (SCI 2003), pp. 30–36. 2003.

[BöJK03] Böhmann, T.; Junginger, M.; Krcmar, H. 2003. Modular Service Architectures: A Concept and Method for Engineering IT Services. Paper presented at the 36th Annual Hawaii International Conference on System Sciences (HICSS-36), January 6-9, 2003, Big Island, Hawaii

[BVEL04] S. Brockmans, R. Volz, A. Eberhart, P. Löffler: Visual Modeling of OWL DL Ontologies Using UML. International Semantic Web Conference 2004: 198-213

[BuSc06] Bullinger, H.J., Scheer A-W. Service Engineering. Springer Verlag, Berlin 2006

[BWFK04] Böhmann, T.; Winkler, T.; Fogl, F.; Krcmar, H.: Servicedatenmanagement für IT-Dienstleistungen: Ansatzpunkte für ein fachkonzeptionelles Referenzmodell. In: Becker, J.;

Delfmann, P. (Hrsg.), Referenzmodellierung: Grundlagen, Techniken und domänenbezogene Anwendung, pp. 99-124. Physica, Heidelberg, 2004.
[Grub95] Gruber, T. R.: Towards principles for the design of ontologies used for knowledge sharing. International Journal of Human-Computer Studies, 43(5/6):907-928, 1995.
[HeRo07] Hepp, M., Roman, D.: An Ontology Framework for Semantic Business Process Management, proceedings of the 8th international conference Wirtschaftsinformatik 2007, February 28 - March 2, 2007, Karlsruhe. In: Oberweis, A; Weinhardt, C.; Gimpel, H.; Koschmider, A.; Pankratius, V.; Schmizler, B.: eOrganisation: Service-, Prozess, Market-Engineering, Vol. 1, Universitaetsverlag Karlsruhe, pp. 423-440
[HLDW05] 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.
[ISO20000] http://www.iso.org/
[ITSM04] IT Service Management Forum: IT Service Management – An Introduction. Van Haren Publishing, Amsterdam, 2004.
[Jabl94] Jablonski, S.: MOBILE: A Modular Workflow Model and Architecture, Proc.of Int'l Working Conference on Dynamic Modelling and Information Systems, Nordwijkerhout, 1994.
[JaBS97] Jablonski, S.; Böhm, M.; Schulze, W.: Workflow-Management: Entwicklung von Systemen und Anwendungen — Facetten einer neuen Technologie. Dpunkt Verlag, Heidelberg, 1997.
[Kicz96] Kiczales, G.: Aspect-oriented programming. ACM Computing Surveys, 28(4), 1996.
[KiFo02] Kim, Henry M. and Fox, Mark S. (2002). "Using Enterprise Reference Models for Automated ISO 9000 Compliance Evaluation", Proceedings of the 35th Hawaii International Conference on Systems Science.
[KlWe01] Ralf Klischewski, Ingrid Wetzel: Modeling Serviceflow. ISTA 2001: 261-272
[KoOb05] Koschmider, A.; Oberweis, A.: Ontology based Business Process Description. In: Proceedings of the CAiSE´05 WORKSHOPS, no. 2, pp. 321-333, Portugal, 2005.
[LeRo97] Leymann, F.; Roller, D.: Workflow-based applications. IBM Systems Journal, 36(1), 1997.
[McHa04] McGuinness, D. L.; Harmelen van, F.: OWL Web Ontology Language - Overview. http://www.w3.org/TR/owl-features, 2004.
[PROT] http://protege.stanford.edu/
[RoGr02] Rosemann, M.; Green, P.: Developing a meta model for the Bunge-Wand-Weber ontological constructs. In: Information Systems (27), pp. 75-91, 2002.
[RoIG04] Rosemann, M.; Indulska, M.; Green, P.: A Reference Methodology for Conducting Ontological Analyses. In: Proceedings of the 23rd International Conference on Conceptual Modelling (ER 2004). pp. 110-121, Shanghai, 2004.
[Sche91] Scheer, A.-W.: ARIS – Modellierungsmethoden, Meta-model le, Anwendungen; Springer Verlag Berlin,1991
[ScBa07] Schmidt, R.; Bartsch C.: Ontology-based Modelling of Service Processes and Services. In: Proceedings of the IADIS International Conference, Applied Computing 2007, pp. 67-74, Salamanca, Spain, 2007.
[Schm03] Schmidt, R.: Web Services Based Architectures to Support Dynamic Inter-organizational Business Processes. International Conference ICWS - Europe 2003-10-09, Erfurt. In: LNCS 2853, Springer, pp. 123-136, 2003.
[Schm06] R. Schmidt: Sercomp: A Component-Oriented Method for Flexible Design and Support of Inter-Organizational Service Processes. To appear in Software Process: Improvement and Practice" (SPIP). Preview available:
[SoHa06] Ljiljana Stojanovic, Hans-Jörg Happel. Ontoprocess - a prototype for semantic business process verification using SWRL rules. 3rd European Semantic Web Conference, Budva, Montenegro, 2006

[SPGK06] Evren Sirin, Bijan Parsia, Bernardo Cuenca Grau, Aditya Kalyanpur and Yarden Katz. Pellet: A practical OWL-DL reasoner, Journal of Web Semantics (To Appear), 2006. (Download)
[SuSt05] Sure, Y.; Studer, R.: Semantic Web Technologies for Digital Libraries. www.aifb.uni-karlsruhe.de/ WBS/ysu/publications/2005_sw_for_dl.pdf, 2005.
[ThFe06] O. Thomas, M. Fellmann: Semantische Ereignisgesteuerte Prozessketten. DW 2006, Friedrichshafen, Germany.
[WeKl03] I. Wetzel, R. Klischewski: Serviceflow Beyond Workflow? Concepts and Architectures for Supporting Inter-organizational Service Processes. CAiSE 2002: 500-515
[WeKl04] I. Wetzel, R. Klischewski: Serviceflow beyond workflow? IT support for managing inter-organizational service processes. Inf. Syst. 29(2): 127-145 (2004)
[ZdHe05] Zdravkovic, J.; Henkel, M..: Enabling Flexible Integration of Business and Technology in Service-based Processes. Proceedings of the CAiSE'05 Workshop, p. 107 – 114.

A Model-driven Approach for Internal Controls Compliance in Business Processes

Kioumars Namiri[1], Nenad Stojanovic[2]

[1] SAP Research Center CEC Karlsruhe, SAP AG, Vincenz-Prießnitz-Str.1
76131 Karlsruhe, Germany
Kioumars.Namiri@sap.com

[2]FZI Karlsruhe, Haid-und-Neu-Str. 10-14
76131 Karlsruhe, Germany
Nenad.Stojanovic@fzi.de

Abstract. Enterprises require mechanisms to ensure that their business processes implement and fulfill internal controls in context of regulatory compliance such as Sarbanes Oxley Act. In this paper we propose an approach for the modeling and implementation of internal controls in business processes. The approach is based on the formal modeling of internal controls, thus it can serve as the basis for usage of logic mechanisms in the compliance verification process.

1 Introduction

The advent of regulatory compliance requirements such as Sarbanes Oxley Act 2002 (SOX)[1] requires the implementation of an effective internal controls system in enterprises. COSO[2] defines the internal controls as a process designed to provide reasonable assurance regarding the achievement of objectives in effectiveness and efficiency of operations, reliability of financial reporting and compliance with applicable laws and regulations. We focus on the Application Controls (AC) controlling business processes and propose the introduction of an abstraction layer above a business process, in which these controls are formally modeled and evaluated against existing process models and instances. We see several advantages of such an approach:

- It enables usage of formal methods for the verification of a business process's compliance.
- Consequently the compliance can be performed automatically based on the current state of a process
- The changes of the controls will not affect the design and execution of the original business processes
- Non-experts can built on top of the domain model provided to design controls for business processes

2 Motivating Scenario

The internal controls compliance of a purchase ordering process (PO) depends on enterprise specific risk assessment carried out by auditing consultants (see Table 1)

Table 1 Risk assessment on Purchase Ordering Process (PO) for an enterprise

Control Objective	Risk	Application Control
Prevent unauthorized use of PO Process	Unauthorized creation of POs and payments for not existing suppliers	Double Approvals of POs higher than $5000 (*Double-Check-Control*).

[1] Pub. L. 107-204. 116 Stat. 754, Sarbanes Oxley Act (2002)

[2] Committee of Sponsoring Organizations of the Treadway Commission (COSO)

3 Domain Model for Internal Controls Compliance

The design of a control should control the way a business process is executed. A (re)design of a business process causes an update of risk assessment on a business process, which may lead to a new or updated set of the controls incl. new tests. The business process monitoring and verification techniques may be used to assess the design of controls and serve as an input to the compliance certification (See Figure 1).

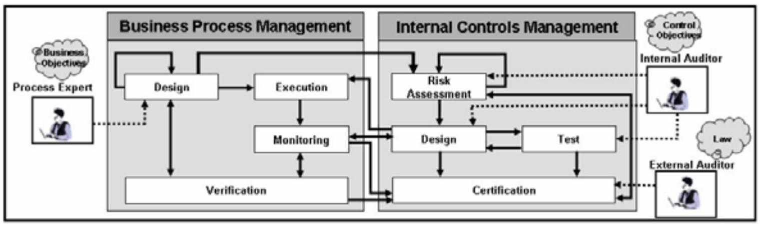

Figure 1 Relations between BPM and Internal Controls Management

The main entities for the process of internal controls compliance is described in following and illustrated in Figure 2a: Identify all **significant accounts** in the company. Identify for those accounts all **business processes** affecting them. Define for each relevant business process a set of **control objectives** specific to the enterprise. Assess the **risks** for the enterprise by their identification for each control objective. Design and implement based on the risk assessment a set of **controls** in order to prevent or detect the occurrence of the identified risks.

An *Application Control* (AC) controls different dimensions of the way a business process is enacted, namely the execution of its *activities*, the *Business Documents* involved and the *agents* performing an *activity* including their *authorities* (See Figure 2b).

For each *AC* at least one *Recovery Action* must have been designed, which reacts on the violation of a control. It does *not* change the designed business process logic; it rather blocks the transaction and may send a notification to an assigned responsible agent.

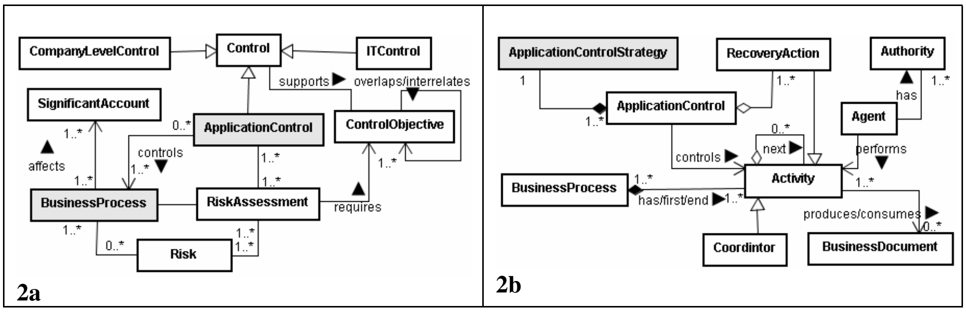

Figure 2a - The upper domain model of the Internal Controls Compliance
Figure 2b - Relationship between an Application Control and a Business Process

Application Control Strategy Model

An *Application Control Strategy* defines the way a control monitors the behavior of one or more activities inside a business process (Figure 3). In order to become active an *AC* requires to be triggered according to the *state* of the process parameters in a *scope*. We define further two elements of an AC strategy: *scope* and *pattern* based conceptually on the work done by Dwyer et al [1]. Although their patterns are mainly used for defining formal requirements on program specifications, they can be applied to internal controls compliance and the monitoring requirements there. For a detailed description of the scopes and patterns and their semantics please refer to [1].

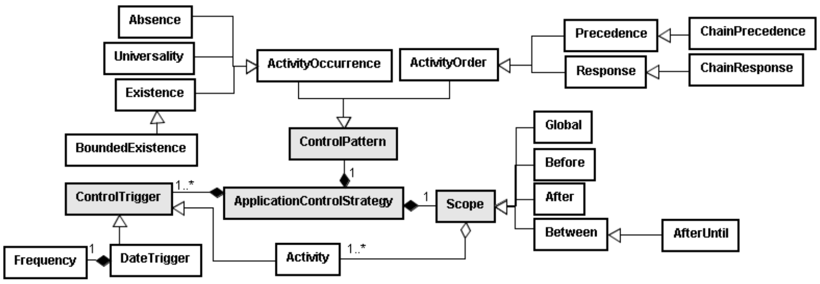

Figure 3 A Semi-formalization of the control implementation

4 The Approach

The abstraction layer above business process model we call the "Semantic Process Mirror" (*SemanticMirror*). According to assessed risks, a set of ACs is defined in this layer. During execution of a business process, this layer will be updated with information needed for the evaluation of defined controls in order to ensure that compliance checks will pass. The approach spans over there phases:

Phase 1: Semantic process mirror design phase
SemanticMirror represents a semantic layer placed on the top of the (usual) syntactical description of a business process (i.e. workflow). In this phase a model of the business process according to Figure 2b will be stored in the SemanticMirror. It will be used later during the phase 2 and 3 to infer whether the process is designed and executed according to a set of declaratively designed ACs in phase 2.

Phase 2: Application control design phase
In the following we present a set of formalizations needed for the automatic evaluation of ACs.
Control statement CS is a logical statement that describes how to carry out an AC *ac* in a business process *bp*:

$$CS(ct, bp, ac(x, cp),GS(bp, scope(M)), \text{action}_R) :=$$
$$O(ct) \wedge V(bp, ac(x, cp), GS(bp, scope)) \rightarrow Activity(bp, \text{action}_R),$$

where the formula for *CS* expresses that if a *violation V* for the given *ac* occurs (is true) after *occurrence O* of a *ControlTrigger ct* on a *Guarded Sequence GS*, then the corresponding *recovery action* action_R will be instantiated and executed on current instance of *bp* (the instance that generated the violation). We describe the parameters mentioned above: *Guarded Sequence* is a sequence of activities, which are along the *scope* of the *AC strategy* of an *ac* in a *bp*. The values for the violation of a control are calculated by evaluating the statement *ac* on the SemanticMirror, i.e. if the statement *ac* can be inferred from the set of facts contained in the SemanticMirror.

An *AC ac* expresses that a *control pattern cp* (See Figure 3) must hold if the logical condition on an entity *x* holds:

$$ac(x, cp) := condition(x) \rightarrow cp, x \in \{BusinessDocument, Agent)$$

We show the formalization of the control pattern (cp) *BoundedExistence* of *n* (see Figure 3) for an *activity C* in the scope of activities defined by *GS*(*bp,scope*)*:*

$$\text{BoundedExistence}(n, C, GS\ (bp, scope)) :=$$

$$(\bigwedge_{i=0,\dots,n} \exists C_i \mid \text{InstanceOf}(C_i, C)) \wedge (\forall_{i,j=0,\dots,n} C_i, C_j \mid C_i != C_j) \wedge (\forall_{i=0,\dots,n}^{n} C_i \mid C_i \in GS\ (bp, scope))$$

Example: Applied on the Double-check control in the PO-Process (see scenario) the statement *ac* looks as follows:

$$\forall PO \mid \text{BusinssDocument}(PO) \wedge \text{Amount}(PO, amount) \wedge \text{greater}(amout, 5000) \rightarrow$$
$$\text{BoundedExistence}(2, \text{ApprovePO}, GS_{\text{DoubleCheck}}(P2P, \text{Between}(\text{SelectSupplier}, \text{SendPO})))$$

Phase 3: Business process execution phase

This phase enables the bidirectional interaction between BPM and internal controls management (see Figure 1): The SemanticMirror will be updated by information about the current instance of the business process enacted and if an AC is violated, the recovery action defined in the control statement will be executed. KBAs represent conceptual abstraction of a log channel, which maybe used to update the SemanticMirror.

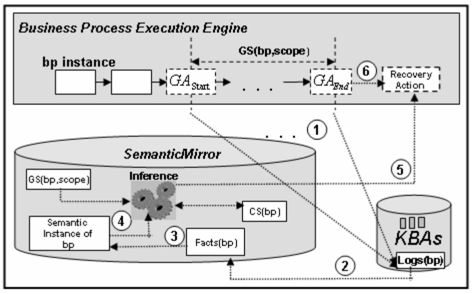

Figure 4 Business process execution phase

5 Related work and conclusion

In this paper we introduced a semantic based approach for conceptual modeling of internal controls required by regulations such as SOX. The controls are captured declaratively and checked during execution-time of business processes. On a conceptual level our work is related to [2], where a taxonomy of risks for business processes is provided. In [3] the logic behind the obligations and permissions on a business process and contracts is made using temporal deontic logic. [4] gives an overview and discusses the current industrial software products in this area and their limitations.

References

1. M. Dwyer, G. Avrunin, J. Corbett, Patterns in Property Specification for Finite-State Verification. In Proceedings of the 21st International Conference on Software Engineering, pages 411-420, May 1999
2. zur Muehlen, Michael; Rosemann, Michael. Integrating Risks in Business Process Models. In: Proceedings of the 2005 Australasian Conference on Information Systems (ACIS 2005), Manly, Sydney, Australia, November 30-December 2, 2005.
3. Guido Governatori, Zoran Milosevic, and Sahzia Sadiq. Compliance checking between business processes and business contracts 10th International Enterprise Distributed Object Computing Conference (EDOC 2006). IEEE Press, 2006, pp. 221-232
4. R. Agrawal, Ch. Johnson, J. Kiernan, F. Leymann: Taming Compliance with Sarbanes-Oxley Internal Controls Using Database Technology. Proc. 22nd Int'l. Conf. on Data Engineering ICDE'2006 (Altanta, GA, USA, April 3 – 7, 2006)

Semantic Business Process Analysis

Irene Celino[1], Ana Karla Alves de Medeiros[2], Gernot Zeissler[3], Michael Oppitz[3], Federico Facca[1], and Stefan Zoeller[3]

[1] CEFRIEL – Politecnico of Milano, Via Fucini 2, 20133 Milano, I
[2] TUE - Technische Universiteit Eindhoven, Postbus 513, 5600 MB, Eindhoven, NL
[3] IBIS Prof. Thome AG - Mergentheimer Str. 76a, 97082 Wuerzburg, D

Abstract. Current Business Process Management technologies cover all the process life-cycle but still suffer from many limitations with respect to their complexity, maintainability and degree of automation. Recent research initiatives aim at overcoming these limitations by introducing Semantic technologies in the process life-cycle. One of the steps that can benefit from this approach is the Business Process Analysis, that focuses on the delicate phase of studying, testing and evaluating existing and running systems and processes, with the aim of identifying the current system (process) state, as well as pointing out problems and bottlenecks, measuring key performance indicators and suggesting potential improvements. We believe that the use of Semantic Web can be of great help in improving and partially automating Business Process Analysis tasks. In this position paper, we explain how we envision the future of Semantic Business Process Analysis and we introduce the early results of our approach based on two different analysis methodologies, Reverse Business Engineering and Process Mining.

1 Introduction

The aim of Business Process Management (BPM) is to manage, support and analyze business operations with a high level managerial perspective. Unfortunately, current technologies on the market are still accessible to IT experts only, because of the gap between the models from the management perspective (e.g. BPMN) and the actually deployed process models (e.g. BPEL). Therefore, business experts still depend on the IT personnel to get feedback about the system. Things get more and more complex when companies merge and have to integrate their processes or when they want to analyze their current processes so as to reengineer them. The major obstacle towards a truly unified view on business processes is that *the business processes inside an organization are widely not accessible to machine reasoning*. All the steps of the BPM life-cycle (Modeling, Configuration, Execution and Analysis) suffer from those problems. Current researches [1] envision the use of Semantic technologies to increase the level of automation in BPM and to overcome the gap between the business experts and the IT people. In particular, this paper explains our research on introducing semantic technologies in the Business Process Analysis (i.e., the study and evaluation of the current running processes with the aim to measure performance, find problems and solve them). The results reported here refer to the activities within the EU-funded project SUPER[1] and adopt the technologies developed within it. The core of our approach is to link the data necessary for the analysis with defined ontological concepts. This linkage lifts the current analysis techniques from a label-based (or string-based) level to a concept-based level and, therefore, enables automatic processing and machine reasoning which, in turn, can help in decreasing the gap between the management and the IT world. Our analysis approach is illustrated by adding semantic to two prominent analysis techniques: Process Mining and Reverse Business Engineering. The idea is to improve the current status of the art that relies on huge manual efforts introducing some degree of automation thanks to the use of semantic annotations in all the steps of the business process life-cycle. This, as discussed in this paper, can result also the improvement of the quality and correctness of the outcome of the business process analysis.

The remainder of this paper is organized as follows. Section 2 introduces the current analysis techniques. Section 3 explains how to add semantic to these techniques. Section 4 contains the conclusions and future steps.

2 Business Process Analysis

Nowadays, companies use various kind of information system (e.g. ERP, CRM, Workflow Management, etc) to support the execution of their business processes. These information systems typically store data about how instances of given processes were executed (i.e. transaction, master and configuration data). These stored data are the starting point for process mining (PM) and reverse business engineering (RBE). Both techniques support the analysis of process models (and processes) and are complementary. PM mainly focuses on *discovery*-like kind of analysis (e.g., How are the processes actually being executed? What is the organizational model for a given process? Where are the bottlenecks in processes?) while RBE targets scenario-based analysis using predefined business questions (e.g., As-Is-Analysis, Continuous Improvement). The rest of this section contains more details about PM and RBE.

2.1 Process Mining

Process mining [2] aims at automatically discovering analysis information about processes. The analysis is based on event logs that contain data about the execution of these processes. The basic assumptions are that (i) an event log should uniquely identify different process instances (or executions) of a given process, and (ii) tasks (or steps) in a process instance are registered in the order in which they were performed. Typical examples are the mining of a model that portray the routes followed by a group of process instances, social networks for the handover of work in a process and the automatic inference of groups and teams for given processes, auditing tools (that need to inspect data fields linked to tasks of process instances) and performance bottleneck analysis (that highlights the specific cases that contain performance issues, the severity of these issues, and so on). Most of the process mining techniques are freely available in the open source tool ProM[2] [3].

[1] SUPER project (FP6-026850): `http://www.ip-super.org`

[2] ProM, `www.processmining.org`.

2.2 Reverse Business Engineering

Reverse Business Engineering (RBE) is a method to analyse productive ERP systems in an automated way. The analysis results comprise transaction usage, expansions, customization, master and transaction data. The two main scenarios are to derive a model of the active and used system elements based on the analysis results (*As-Is-Analysis*) and to gain information about gaps, exceptions or potentials within an ERP system in order to redesign and improve the business processes and the underlying ERP system (*Continuous Improvement*).

The methods behind Reverse Business Engineering (RBE) were developed at the University of Wuerzburg and applied to the SAP R/3 System by IBIS Prof. Thome AG in collaboration with SAP AG[3] [4].

3 Introducing "Semantics" into the BP Analysis

Although process mining and reverse business engineering techniques provide feedback about different perspectives of process models, the degree of automation and reuse is somewhat limited because it is based on strings in event logs (for PM) or raw data in ERP (for RBE). So, this section gives an overview on how the use of semantic can improve the analysis supported by PM and RBE techniques by bringing them to the concept-level. Our approach towards a semantic analysis environment has three steps: (1) the *creation of ontologies* that capture the meanings of different elements (tasks, data fields, performers, etc.) in process models, (2) the *semantic annotation of business processes* with the defined ontologies, and (3) the definition of *semantic versions* of existing PM and RBE techniques. For what regards the *ontologization of data*, we are defining an ontology framework (in line with [5]) that comprises the relevant concepts for events description and business questions formulation. Then, we will use these ontologies to annotate the processes, by mapping (possibly in a semi-automated way) at design time the business questions onto the processes, tasks and data fields they refer to, and by assuring that the execution logs will contain the references to those ontology concepts. For what regards the *ontologization of techniques and tools*, as explained in Section 2, PM mainly focuses on discovery-like analysis and RBE on scenario-based one using predefined business questions. Thus, here we propose five possible semantic extensions for these techniques: semantic process discovery, semantic organizational model discovery, semantic auditing, semantic performance analysis and semantic RBE. *Semantic Process Discovery* builds hierarchical models based on subsumption trees for the ontologies in event logs, while current process mining techniques only capture a flat representation of process models. *Semantic Organizational Model Discovery* automatically discovers groups and teams in organizations, based on task similarity. The current version of these techniques uses string matching as the criterion to assess task similarity: the linkage of ontological concepts to task will allow for smarter inferences of tasks similarities. *Semantic Auditing* will allow the validation properties to be defined in terms of (sub-)concepts in a log. Currently, this auditing is based on strings in the log, what greatly hinders the re-use of

[3] RBE PlusTM, `http://www.ibis-thome.com/rbe`

defined properties and make the definition of these properties too technical (recall the gap between the management and the IT world). *Semantic Performance Analysis* will use the semantic annotations to automatically identify bottlenecks in the system and violations of service level agreements. At the present time, these techniques do not have the notion of what an acceptable execution time would be for certain tasks or processes: the defined ontologies can capture these notions. The focus of *Semantic RBE* is the business question ontology which allows the generalization and reusability of RBE content and provides semantic analysis. Thus a flexible and standardized adoption of RBE to various kind of application systems, process models and respective modelling solutions can be realized.

4 Conclusions and Future Work

This paper shows how current business process analysis techniques can benefit from the use of semantic information. The main idea is to annotate the elements that are relevant for analysis with ontological concepts. The benefits are two-fold: (i) by using ontologies and, therefore, performing analysis at the concept-level, the proposed solutions reduce the gap between the management and the IT worlds in companies, and (ii) the use of ontologies greatly promote the reuse of analysis queries etc. Our future work will consist on defining the ontologies for analysis purposes and on implementing the five semantic extensions proposed in this paper. This future work is part of the SUPER European project.

Acknowledgments

We would like to thank John Domingue, Emanuele Della Valle, Carlos Pedrinaci, Wil van der Aalst and the whole SUPER Consortium for their valuable support and contribution.

References

1. 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. In: Proceedings of the IEEE international Conference on E-Business Engineering. (2005)
2. van der Aalst, W., Weijters, A.: Process Mining. In: Process-Aware Information Systems: Bridging People and Software through Process Technology. Wiley & Sons (2005) 235–255
3. van Dongen, B., de Medeiros, A., Verbeek, H., Weijters, A., van der Aalst, W.: The prom framework: A new era in process mining tool support. In Ciardo, G., Darondeau, P., eds.: Proceedings of the 26th International Conference on Applications and Theory of Petri Nets, LNCS 3536 (2005) 444–454
4. Wenzel, H.: Reverse business engineering: Ableitung von betriebswirtschaftlichen modellen aus produktiven softwarebibliotheken. Published dissertation at the chair of business administration, University Wuerzburg (1999)
5. Hepp, M., Roman, D.: An Ontology Framework for Semantic Business Process Management. In: Proceedings of Wirtschaftsinformatik 2007. (2007)

KISS – Knowledge-intensive Service Support for Agile Process Management

Daniela Feldkamp, Knut Hinkelmann, Barbara Thönssen

University of Applied Science Northwestern Switzerland, School of Business
Riggenbachstr. 16, 4600 Olten, Switzerland
{daniela.feldkamp, knut.hinkelmann, barbara.thoenssen}@fhwn.ch

Abstract. Automating business processes especially in the tertiary sector is still a challenge as they are normally knowledge intensive, little automated but compliance relevant. To meet these requirements the paper at hand introduces the KISS approach: modeling knowledge intensive services by enriching business rules semantically and linking rules to processes.

Keywords: Variable Process, Business process management, Business rules.

1 Introduction

Business process management has been very successful for structured processes with the objectives of process optimization, quality management, implementation of business information systems, or workflow management. In actual applications, however, we still face various problems: Often process documentations are not in line with the real work in the organization, e.g. because the processes are not implemented as documented or because processes have changed and the documentation is not adjusted. Also, process definition often lack the right level of granularity, i.e. they are very detailed forcing participants to follow a rigid regime and prohibiting flexibility in process execution. The situation is even worse for knowledge-intensive or dynamic processes as they have to deal with exceptional situations, unforeseeable events, unpredictable situations, high variability and highly complex tasks.

As a consequence, knowledge intensive processes are weakly structured and do not match at least the one crucial condition for process automation: A high repeatability rate, i.e. doing the same thing in the same way many times. Consider for example approving a building application, it may be necessary to conduct several checks including inspection on location, approval of application by historical preservation agency, and assessment of environmental compatibility. All tasks depend on each other. However the outcome of one task also may be that the application will fail; then no further tests are required. Dietz [2] states, that some modeling approaches treat 'information processes' like 'real business processes' (as Dietz calls it) therefore failing to meet the requirements. If possible at all such a process model, covering all possible cases, would be highly complex and difficult to manage. Because, of this knowledge intensive processes are normally not automated and often not even documented in detail. The disadvantages of that situation are lack of transparency and traceability of work, inconsistent decisions and neglected company's regulations. Especially the increasing demands on governance and compliance have been forcing companies as well as public administrations in the last few years to review these kinds of processes for improvement.

That brings the subject of 'Business Rules' into the picture. Business rules allow for an explicit and consistent specification of regulations [5]. They provide an excellent means of encapsulating knowledge about the intentions of business and their implementation in business processes and information systems [4].

2 The KISS Approach

The KISS approach combines business rules and process models in order to automate knowledge intensive services by taking advantage of both fields:

- process models are used for
  - explicit documentation and visualization
  - execution automation.
- rules are used for
  - variable process execution: determine activities and processes to be executed taking into account for dependencies between activities
  - Intelligent resource allocation at run time: selection of employees based on special skills and selection of particular web services adequate for the actual circumstances
  - Intelligent branching and decision making: deriving workflow-relevant data using inferences and computing values
  - Consistency checking: avoiding violation of integrity constraints and guidelines.

This means, that an activity of a process can have relations to four different rule sets, one rule set for each of the above mentioned relation types. To allow for flexible process execution we introduced a new modelling construct that we call *variable activity*. A variable activity is closely related to a knowledge-intensive task as introduced by Abecker et al. [1]. It corresponds to a subprocess with the particularity that the activities of this kind of subprocess are determined at run-time instead of strictly modelled at build time. At run-time rules select the activities that have to be executed depending on the actual context of the process instance, allowing for resource allocation and supporting the user in decision making, while integrity checking based on constraints and guidelines (in combination with inference rules) ensure consistency and compliance.

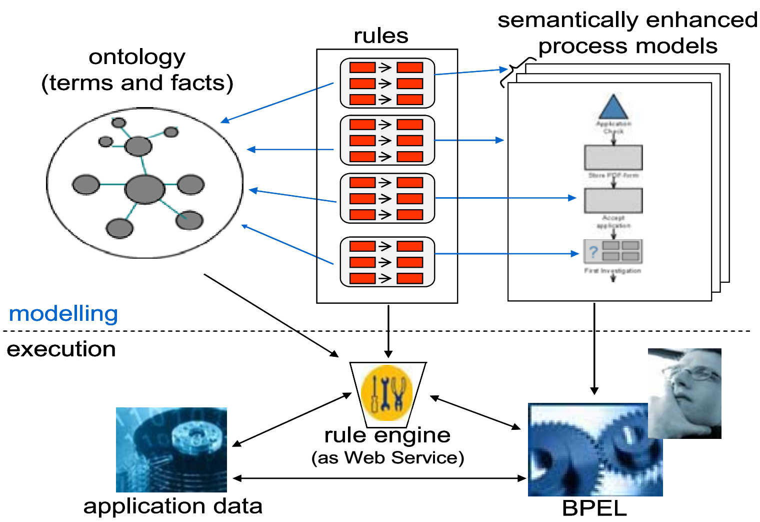

Fig. 1 Extension of traditional process modeling and execution

Therefore, we partly agree to Hepp and Roman [3] stating that explicitly modelling the process flow has several disadvantages hence preferring a declarative process description instead. However, as certain business constraints *do* cause 'fixed' dependencies on activities (e.g. sequence: a claim must always be verified before it can be decided on), the KISS approach suggests to model these parts as a flow in a process model as it is much better to visualize and to understand as purely declarative descriptions.

The KISS approach has been developed in the FIT project[1] and is applied for e-government services of the Austrian city of Voecklabruck. Although there are binding legal rules and regulations every administration has to obey, dealing with people's concerns means dealing with different circumstances every time. In this sense, e-government services are often knowledge intensive processes, where the actual process execution and the involved participants and administrations depend on various factors.

Ontologies build the basis for modelling and executing semantically enriched processes and business rules. The vocabulary of rules is represented in an OWL ontology, which can base on existing ontologies, like ontologies for the modelling of commercial and public enterprises provided by the TOVE project[2]. The rules are represented using SWRL. Semantically enhanced process models (represented in OWL-S) allow for context-dependent invocation of the business rules. Using ontologies for the representation of facts and terms has the advantage of higher expressiveness and the chance to use inferences like inheritance and consistency checking.

---

[1] FIT (Fostering self-adaptive e-government service improvement using semantic Technologies) is a project funded by the European Commission within the IST programme, IST-2004-27090

[2] http://www.eil.utoronto.ca/enterprise-modelling/tove/index.html.

3 Conclusion

The KISS approach for agile process management leads to more flexible and agile business processes by integrating business process management and business rules in four different ways. With the KISS approach, process models can be kept small and simple (following the slogan "KISS = Keep It Simple and Small") representing only the mandatory process flow while the knowledge is separated in the business rules. This has the advantages, that rules can be modified independently from the business logic. Additionally, one rule can relate to several activities, so the business logic is more reusable.

References

1. Abecker, A.; Bernardi, A.; Hinkelmann, K.; Kühn, O.; Sintek, M. (1998): Toward a Well-Founded Technology for Organizational Memories. IEEE Intelligent Systems and their Applications, Vol. 13, No 3, p. 40 - 48, May/June 1998.
2. Dietz, J.L.G., The deep structure of business processes, Communications of the ACM, Volume 49 , Issue 5 (May 2006), pp. 58 – 64.
3. Hepp, M. and Roman, D. (2007): An Ontology Framework for Semantic Business Process Management, Proceedings of Wirtschaftsinformatik 2007, February 28 – March 2, , Karlsruhe.
4. Morgan, T. (2002): Business Rules and Information Systems: Aligning IT with Business Goals. Addison-Wesley, Boston.
5. Ross, R. G. (1998): Business Rules Concepts - Getting to the Point of Knowledge. Business Rule Solutions, LLC, Houston.

An Ontology for Executable Business Processes

Jörg Nitzsche, Daniel Wutke, and Tammo van Lessen

Institute of Architecture of Application Systems
University of Stuttgart
Universitaetsstrasse 38, 70569 Stuttgart, Germany
{joerg.nitzsche,daniel.wutke,tammo.van.lessen}
@iaas.uni-stuttgart.de
http://www.iaas.uni-stuttgart.de

Abstract The *Web Service Business Process Execution Language (WS-BPEL)* is the de facto standard for describing workflow-like compositions of Web services, so-called Web service orchestrations. In this paper an ontology for executable BPEL processes is presented, which reflects both the natural language description and the syntax given in the specification. The ontology makes BPEL process models accessible at a semantic level and thus to intelligent queries and machine reasoning.

Key words: BPM, BPEL, ontology, semantics, WSML

1 Introduction

An ontology is an explicit formal specification of a domain [1]. It consists of (i) a number of concepts represented as *classes* supporting the definition of hierarchies through (multiple) inheritance, (ii) *instances* of concepts representing concrete objects of the ontology, (iii) *relations* between concepts, and (iv) *axioms* that capture knowledge that cannot be inferred. An ontology is described using a formally defined language.

This document presents an ontology for the *Web Service Business Process Execution Language 2.0 (WS-BPEL)* [2] using the formalism of the *Web Service Modeling Language (WSML)* [3].

WS-BPEL (BPEL for short) is the de facto standard for describing Web service flows in a workflow-like manner by combining activities that represent interaction with Web services (`invoke`, `receive`, `pick`, `reply`) with control flow activities (`flow`, `sequence`, `while`). BPEL [2] has been approved by the WS-BPEL Technical Committee as a Committee Specification in 01/2007. It enables the composition of Web services [4] and the modeled process is itself exposed as a Web service. Thus it provides a recursive aggregation model for Web services.

BPEL is the foundation of process execution and represents the IT view on processes in Semantic Business Process Management (SBPM) [5] which aims at bridging the gap between the business and IT view. Building an ontology of BPEL and representing BPEL processes ontologically facilitates reasoning about executable process models. Through semantic annotation of BPEL activities (e.g. relating the concept of a service invocation to a concept that represents the purpose of the invocation), queries can be formulated on the semantics of a process, rather than its syntactic representation. Ontologies of business centric process notations such as BPMN [6] and EPC [7] together with the BPEL ontology make up the orchestration part of the process space defined in [8]. Using ontology mediation queries on the business level (BPMN, EPC) can involve reasoning BPEL processes, i.e. queries can span multiple levels of abstraction. Additionally, the semantically enriched audit log of a BPEL engine in conjunction with the process models expressed in terms of the BPEL ontology can be used for semantic mining, i.e. reasoning not only on the process model level but also on the process instance level.

WSML [3] is a family of Web ontology languages with native support for the conceptual framework of the *Web Service Modeling Ontology (WSMO)* [9]. The main features of the language are support for modeling classes, attributes, binary relations and instances. Additionally, the language supports class hierarchies. WSML has multiple variants, possessing different levels of logical expressiveness and the use of different language paradigms, providing its users with the possibility to choose between the provided expressivity and the implied complexity. WSML Core forms the foundation of all the WSML variants, providing the minimum expressivity through function- and negation-free Datalog [3]. WSML Core is extended in two directions: description logics (WSML-DL) and logic programming (WSML-Flight and WSML-Rule). The variant WSML-Full unifies both extensions and therefore allows the creation of ontologies which are undecidable.

For the BPEL ontology WSML-Flight was chosen since it is the least expressive variant that allows for cardinality constraints and inequality in the logical language, which are necessary for the definition of the ontology. The ontology was defined following the basic principles of ontology development, as described in [10].

2 Building the ontology

The conceptualization of BPEL is based on the XML schema defined by the BPEL specification. The general rule we follow is to express each BPEL XML element as a class in the BPEL ontology and its corresponding attributes as attributes of the class. When possible, WSML built-in data types were reused as attributes (e.g. the strings “yes” and “no” are represented as values of type boolean rather than string). In case of conflicting names for concepts defined in the ontology and WSML keywords, the concept names are represented as full IRIs [11]. Cardinality constraints defined in the XML schema definition were adopted as far as possible. This initial conceptualization is enriched by a formal description of information described in the specification in natural-language form and not expressed in the XML schema definition. Due to this semantic enrichment of the definition of the BPEL syntax, certain aspects of the specification (e.g. the concept of fault handlers, described in Section 2.1 were modeled differently to what their XML schema definition suggests, resulting in an abstraction from the concrete specification of the language syntax.

In the following sections, the BPEL ontology is presented by means of examples of notable design choices.

2.1 Abstraction from XSD

By following the generic approach of ontologizing the XML schema definition of BPEL, unnecessary containers which do not represent any additional semantics are introduced. The `faultHandler` element for instance is only a container that allows specifying multiple `catch` elements and one `catchAll` element to be used within a process or a scope. As the BPEL ontology captures only the semantics of the language and not syntax specific details, it abstracts from these, modeling the concepts Catch and CatchAll (which both inherit from FaultHandler) directly as attributes of the concept Process (Listing 1).

```
concept Process
    nonFunctionalProperties
        xmlns hasValue "http://docs.oasis−open.org/wsBPEL/2.0/Process/executable"
        dc#description hasValue "Concept of being a <process>−Element of an executable BPEL Process"

    endNonFunctionalProperties
    [...]
    hasCatch ofType Catch
    hasCatchAll ofType (0 1) CatchAll
    hasActivity ofType (1) Activity

concept FaultHandler
    nonFunctionalProperties
        dc#description hasValue "Concept of being a <faultHandler>−Element"
    endNonFunctionalProperties
    hasActivity ofType (1) Activity

concept Catch subConceptOf FaultHandler
    nonFunctionalProperties
        dc#description hasValue "Concept of being a <catch>−Element"
    endNonFunctionalProperties
    hasFaultName ofType (0 1) _sqname
    hasFaultVariable ofType (0 1) Variable
    hasFaultType ofType (0 1) DataType

concept CatchAll subConceptOf FaultHandler
    nonFunctionalProperties
        dc#description hasValue "Concept of being a <catchAll>−Element"
    endNonFunctionalProperties
```

Listing 1. Abstraction from XML Schema: Fault Handlers

In some cases however, the XML syntax directly influences the semantics of the BPEL specification. Thus an abstraction is impossible. An example for this scenario is the extension mechanism that enables introducing new activity types (Listing 2). The `extensionActivity` element represents an activity that can be used everywhere in a process where an activity is required. However, it does not come with the standard attributes an activity has to provide, but is a container for a new activity type that has to come with the standard attributes (also depicted in Listing 2).

```
concept ExtensionActivity subConceptOf BasicActivity
  nonFunctionalProperties
        dc#description hasValue "Concept of being an ExtensionActivity, i.e. by inheriting from this activity
                type new operational semantics can be defined"
    endNonFunctionalProperties
    hasActivtity ofType (1) NewActivityType

concept standardAttributes
    nonFunctionalProperties
        dc#description hasValue "Concept of providing the standard attributes For Activities in BPEL"
    endNonFunctionalProperties
    hasName ofType (0 1) _string
    hasSuppressJoinFailure ofType (0 1) _boolean
    isTarget ofType Link
    hasJoinCondition ofType (0 1) Condition
    isSource ofType (0 *) Link
```

Listing 2. Abstraction from XML Schema: Extension activity

2.2 Introduction of hierarchies

In BPEL there are two kinds of activities, basic activities (e.g. `receive`, `reply`, `invoke` and `assign`) and structured activities (e.g. `sequence`, `flow` and `while`). While this information is part of the textual description of BPEL, it is not reflected in the XML schema definition of the language syntax. This kind of knowledge can be added by introducing additional hierarchies. Listing 3 illustrates the activity hierarchy: both BasicActivity and StructuredActivity inherit from the concept Activity. The concepts representing the BPEL elements `receive`, `reply`, `invoke` or `assign` then inherit from BasicActivity whereas the concepts representing `sequence`, `flow` or `while` inherit from the concept StructuredActivity.

```
concept Activity
    nonFunctionalProperties
        dc#description hasValue "Concept of being a BPEL Activity"
    endNonFunctionalProperties

concept BasicActivity subConceptOf Activity
    nonFunctionalProperties
        dc#description hasValue "Concept of being a basic activity"
    endNonFunctionalProperties

concept StructuredActivity subConceptOf Activity
    nonFunctionalProperties
        dc#description hasValue "Concept of being a structured activity"
    endNonFunctionalProperties

concept Assign subConceptOf {BasicActivity, StandardAttributes}
    nonFunctionalProperties
        dc#description hasValue "Concept of being an <assign>−Activity"
    endNonFunctionalProperties
    hasValidate ofType (0 1) _boolean
    hasAssignOperation impliesType (1 *) AssignOperation

concept Sequence subConceptOf {StructuredActivity, StandardAttributes}
    nonFunctionalProperties
        dc#description hasValue "Concept of being a <sequence>−Activity"
    endNonFunctionalProperties
    hasOrderedActivity ofType (1) OrderedActivity
```

Listing 3. Introduction of hierarchies: Activity hierarchy

The multiple inheritance of Assign and Sequence is further elaborated in Section 2.4.

2.3 Axioms

Axioms add semantics to an ontology because they are statements that are assumed to be true without any proof. In BPEL processes communicate with other services on a `partnerLink` which specifies which role the partner service and the process itself take. The BPEL specification informally defines, that a `partnerLink` has to specify at least one role, "myRole" or "partnerRole", which is expressed using the axioms presented in Listing 4.

```
concept _"http://www.ip−super.org/ontologies/sBPEL/20070404#PartnerLink"
      nonFunctionalProperties
            dc#description hasValue "Concept of being a <partnerLink>−Element"
      endNonFunctionalProperties
      hasName ofType (1) _string
      hasPartnerLinkType ofType (1) wsdlx#PartnerLinkType
      hasMyRole ofType (0 1) wsdlx#Role
      hasPartnerRole ofType (0 1) wsdlx#Role
      doesInitializePartnerRole ofType (0 1) _boolean

concept WellformedPartnerlink

axiom definedBy
  ?x memberOf WellformedPartnerLink :− ?x[hasMyRole hasValue ?y] memberOf
PartnerLink.

axiom definedBy
  ?x memberOf WellformedPartnerLink :− ?x[hasPartnerRole hasValue ?y]
memberOf PartnerLink.

axiom definedBy
  !− ?x memberOf PartnerLink and naf (?x memberOf WellformedPartnerLink).
```

Listing 4. Axiom: Well-formed partner link

2.4 Multiple inheritance

A workflow comprises three dimensions: process logic (what is to be done?), organization (who is supposed to do it?), and infrastructure (using which resources?) [12]. In contrast to this, a Web service flow is characterized by only the dimensions process logic, describing the control-flow of a process, i.e. what the process does, and infrastructure, defining the services that implement the activities. Since there are activities in BPEL that deal with both dimensions (control flow and interaction), they inherit from both the interaction- and the control-flow domain. Due to the BPEL extension mechanism, the control-flow domain is split into the Activity branch and the StandardAttributes branch. The chosen design approach of multiple inheritance is depicted in Listing 5. The concept Invoke inherits from WSDLInteraction, BasicActivity and StandardAttributes.

```
concept Interaction
      nonFunctionalProperties
            dc#description hasValue "Concept of interaction"
      endNonFunctionalProperties
```

```
concept WSDLInteraction subConceptOf Interaction
    nonFunctionalProperties
        dc#description hasValue "Concept of doing WSDL dependant interaction"
    endNonFunctionalProperties
    hasPartnerLink ofType (1) _"http://www.ip−super.org/ontologies/sBPEL/20070404#PartnerLink"
    hasPortType ofType (0 1) _sqname
    hasOperation ofType (1) _sqname
    hasCorrelation ofType Correlation

concept Invoke subConceptOf {WSDLInteraction, BasicActivity, standardAttributes}
    nonFunctionalProperties
        dc#description hasValue "Concept of being a <invoke>−Activity"
    endNonFunctionalProperties
    hasCorrelation ofType Correlation
    hasInputVariable ofType (1) Variable
    hasOutputVariable ofType (0 1) Variable
    hasCatch ofType Catch
    hasCatchAll ofType (0 1) CatchAll
    hasPattern ofType (0 1) _string
    hasCompensationHandler ofType (0 1) CompensationHandler
    hasToParts ofType ToParts
    hasFromParts ofType FromParts
```

Listing 5. Multiple Inheritance: Invoke activity

2.5 Workaround for missing expressivity in WSML

In WSML there is no construct that allows expressing the order of elements. Since the BPEL element `sequence` defines a list of activities which are to be executed sequentially and thus requires ordering of elements, the BPEL ontology implements a workaround for this issue. The approach chosen is to model a linked list, as modeling an array would require an axiom to ensure well-formedness by defining that there is only one entry for each position of the array. This is depicted in Listing 6. The activity Sequence has one attribute OrderedActivity which is a container with an attribute Activity and optionally another OrderedActivity. This workaround enables ordering elements. It does not affect the operational semantics of BPEL which are implicit.

```
concept Sequence subConceptOf {StructuredActivity, standardAttributes}
    nonFunctionalProperties
        dc#description hasValue "Concept of being a <sequence>−Activity"
    endNonFunctionalProperties
    hasOrderedActivity ofType (1) OrderedActivity

concept OrderedActivity
    nonFunctionalProperties
        dc#description hasValue "concept of being an item of an ordered Activity list"
    endNonFunctionalProperties
    hasActivity ofType (1) Activity
    hasOrderedActivity ofType (0 1) OrderedActivity
```

Listing 6. Ordered List of Activities

The same approach was used for modeling `elseif` branches which also require ordering (OrderedConditionalBranch).

3 Sample process

By means of a Virtual travel agency process (Figure 1) we show the applicability of the ontology to represent business processes. The travel booking process receives a request, prepares the input data for Web service invocations, invokes the HotelBooking and the FlightBooking Web service, aggregates the result and sends the result back to the requester. The process model in terms of instances of the BPEL ontology is in the appendix. Using this representation, simple queries like "Which activities deal with hotel booking?" can be answered. As invoke is sub concept of basicActivity which is sub concept of activity, it can be inferred that an invoke is an activity. Compared to e.g. a relational databases where a user has to explicitly encode the knowledge that invoke, receive and reply are activities within the query, this knowledge can be inferred by the reasoner when using the BPEL ontology. In conjunction with the process instance data, queries like "Which process instances deal with journeys to Germany or Austria?" can be answered. Assuming appropriate domain ontologies are used to describe the destinations it can be inferred for instance that Berlin and Stuttgart are cities in Germany and Innsbruck and Vienna are located in Austria and are thus included in the result set of the query.

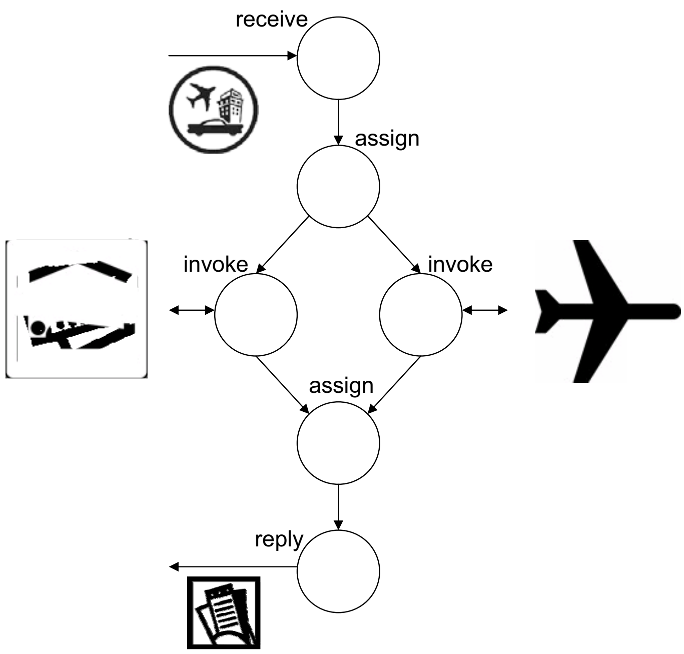

Figure 1. Virtual Travel Agency Process

4 Conclusions

In this document, an ontology of the BPEL specification was presented and the most essential design decisions were explained. Due to the chosen formalism, not all constraints given by the BPEL language can be expressed, i.e. the correctness of a BPEL process can not be checked using the ontology. However, this does not affect the application of the ontology for the purpose of process reasoning: (i) reasoning about process models, (ii) reasoning across different levels of abstraction, i.e. across modelling notations and languages on the business level and the IT level and (iii) reasoning including process instance data.

5 Acknowledgments

We would like to thank Axel Polleres for his helpful comments and explanations regarding the definition of the WSML axioms used in the ontology and Barry Norton for fruitful discussion about the subject matter.

The work published in this article was partially funded by the SUPER project[1] (contract no. FP6-026850) and the TripCom project[2] (contract no. FP6-027324) under the EU 6th Framework Programme Information Society Technologies.

References

1. Gruber, T.R.: Towards Principles for the Design of Ontologies Used for Knowledge Sharing. In Guarino, N., Poli, R., eds.: Formal Ontology in Conceptual Analysis and Knowledge Representation, Deventer, The Netherlands, Kluwer Academic Publishers (1993)
2. Alves, A., Arkin, A., Askary, S., Barreto, C., Bloch, B., Curbera, F., Ford, M., Goland, Y., Guízar, A., Kartha, N., Liu, C.K., Khalaf, R., König, D., Marin, M., Mehta, V., Thatte, S., van der Rijn, D., Yendluri, P., Yiu, A.: Web services business process execution language version 2.0. Committee specification, OASIS Web Services Business Process Execution Language (WSBPEL) TC (January 2007)
3. de Bruijn, J., Lausen, H., Krummenacher, R., Polleres, A., Predoiu, L., Kifer, M., Fensel, D.: D16. 1v0. 2 The Web Service Modeling Language WSML. WSML Final Draft March **20** (2005)
4. 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 Upper Saddle River, NJ, USA (2005)
5. 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. IEEE International Conference on e-Business Engineering (ICEBE 2005). Beijing, China (2005) 535–540
6. White, S.: Business Process Modeling Notation (BPMN) Version 1.0. Business Process Management Initiative, BPMI. org, May (2004)

[1] `http://www.ip-super.org/`
[2] `http://www.tripcom.org`

7. Hoffmann, W., Kirsch, J., Scheer, A.: Modellierung mit ereignisgesteuerten Prozessketten:(methodenhandbuch, Stand: Dezember 1992). Iwi (1993)
8. Hepp, M., Roman, D.: An Ontology Framework for Semantic Business Process Management. In: eOrganization: Service-, Prozess, Market- Engeneering. Volume 1., 8th international conference Wirtschaftsinformatik. Karlsruhe, Germany (2007) 423–440
9. Lausen, H., Polleres, A., Roman, D.: Web Service Modeling Ontology (WSMO). W3C Member Submission **3** (2005)
10. Noy, N., McGuinness, D.: Ontology Development 101: A Guide to Creating Your First Ontology. Knowledge Systems Laboratory, March (2001)
11. Duerst, M., Suignard, M.: Internationalized Resource Identifiers (IRIs). (2005)
12. Leymann, F., Roller, D.: Production workflow. Prentice Hall (2000)

Appendix

```
wsmlVariant _"http://www.wsmo.org/wsml/wsml-syntax/wsml-flight"
namespace { _"http://www.ip-super.org/ontologies/sBPEL/TravelBookingProcess#",
  sbpel _"http://www.ip-super.org/ontologies/sBPEL/20070404#",
  dc _"http://purl.org/dc/elements/1.1#" }

ontology sbpelProcess

     nonFunctionalProperties
          dc#title hasValue "Travel Booking sBPEL Process"
          dc#creator hasValue "Joerg Nitzsche"
          dc#publisher hasValue "SUPER European Integrated Project"
          dc#subject hasValue {"sBPEL", "business process", "workflow"}
          dc#language hasValue "en-UK"
          dc#date hasValue "$Date: 2007/04/04$"
     endNonFunctionalProperties

     importsOntology
          { _"http://www.ip-super.org/ontologies/sBPEL/20070404#",
          _"http://www.ip-super.org/ontologies/wsdlExtension4BPEL/20070126#" }

instance processTravelBooking memberOf Process
  hasName hasValue "ContentProvision"
  hasTargetNamespace hasValue "http://www.ip-super.org/ontologies/prereview"
  hasPartnerLink hasValue travelBookingPL, hotelBookingPL, flightBookingPL}
  hasActivity hasValue processFlow
  hasVariable hasValue {varTravelBookingRequest, varHotelBookingRequest, varFlightBookingRequest,
       varTravelBookingResponse, varHotelBookingResponse, varFlightBookingResponse}

instance typeTravelBookingRequest memberOf MessageType
  hasMessageType hasValue "http://ip-super.org/processes/TravelAgency.wsdl#
       travelBookingRequestMessage"

instance varTravelBookingRequest memberOf Variable
  hasName hasValue "varTravelBookingRequest"
  hasDataType hasValue typeTravelBookingRequest

instance typeTravelBookingResponse memberOf MessageType
  hasMessageType hasValue "http://ip-super.org/processes/TravelAgency.wsdl#
       travelBookingResponseMessage"

instance varTravelBookingResponset memberOf Variable
  hasName hasValue "varTravelBookingResponse"
  hasDataType hasValue typeTravelBookingResponse

instance typeHotelBookingRequest memberOf MessageType
  hasMessageType hasValue "http://ip-super.org/processes/HotelService.wsdl#
```

```
    hotelBookingRequestMessage"

instance varHotelBookingRequest memberOf Variable
  hasName hasValue "varHotelBookingRequest"
  hasDataType hasValue typeHotelBookingRequest

instance typeHotelBookingResponse memberOf MessageType
  hasMessageType hasValue "http://ip−super.org/processes/HotelService.wsdl#
      hotelBookingResponseMessage"

instance varHotelBookingResponset memberOf Variable
  hasName hasValue "varHotelBookingResponse"
  hasDataType hasValue typeHotelBookingResponse

instance typeFlightBookingRequest memberOf MessageType
  hasMessageType hasValue "http://ip−super.org/processes/FlightService.wsdl#
      flightBookingRequestMessage"

instance varFlightBookingRequest memberOf Variable
  hasName hasValue "varFlightBookingRequest"
  hasDataType hasValue typeFlightBookingRequest

instance typeFlightBookingResponse memberOf MessageType
  hasMessageType hasValue "http://ip−super.org/processes/FlightService.wsdl#
      flightBookingResponseMessage"

instance varFlightBookingResponset memberOf Variable
  hasName hasValue "varFlightBookingResponse"
  hasDataType hasValue typeFlightBookingResponse

instance travelBookingPL memberOf _"http://www.ip−super.org/ontologies/sBPEL/20070308#PartnerLink"
  hasName hasValue "travelBookingPL"
  hasPartnerLinkType hasValue travelBookingPLT
  hasMyRole hasValue provider

instance travelBookingPLT memberOf PartnerLinkType
  hasName hasValue "travelBookingPLT"
  hasRole hasValue provider

instance provider memberOf Role
  hasName hasValue "provider"
  hasPortType hasValue "http://ip−super.org/processes/TravelAgency.wsdl#TravelBookingPortType"

instance hotelBookingPL memberOf _"http://www.ip−super.org/ontologies/sBPEL/20070308#PartnerLink"
  hasName hasValue "hotelBookingPL"
  hasPartnerLinkType hasValue hotelBookingPLT
  hasPartnerRole hasValue provider

instance hotelBookingPLT memberOf PartnerLinkType
  hasName hasValue "hotelBookingPLT"
  hasRole hasValue provider

instance provider memberOf Role
  hasName hasValue "provider"
  hasPortType hasValue "http://ip−super.org/processes/HotelService.wsdl#HotelBookingPortType"

instance flightBookingPL memberOf _"http://www.ip−super.org/ontologies/sBPEL/20070308#PartnerLink"
  hasName hasValue "flightBookingPL"
  hasPartnerLinkType hasValue flightBookingPLT
  hasPartnerRole hasValue provider

instance flightBookingPLT memberOf PartnerLinkType
  hasName hasValue "flightBookingPLT"
  hasRole hasValue provider

instance provider memberOf Role
  hasName hasValue "provider"
  hasPortType hasValue "http://ip−super.org/processes/FightService.wsdl#FlightBookingPortType"
```

```
instance processFlow memberOf Flow
  hasActivity hasValue {recTravelBookingRequest, assSplitRequest, invHotelService, invFlightService,
        assMergeResults, repTravelBookingResponse}
  hasLink hasValue {receiveToSplitRequest, splitRequestToInvHotelService, splitRequestToInvFlightService,
        invHotelServiceToMergeResults, invFlightlServiceToMergeResults, mergeResultsToReply}

instance receiveToSplitRequest memberOf Link
  hasName hasValue "receiveToSplitRequest"

instance splitRequestToInvHotelService memberOf Link
  hasName hasValue "splitRequestToInvHotelService"

instance splitRequestToInvFlightService memberOf Link
  hasName hasValue "splitRequestToInvFlightService"

instance invHotelServiceToMergeResults memberOf Link
  hasName hasValue "invHotelServiceToMergeResults"

instance invFlightlServiceToMergeResults memberOf Link
  hasName hasValue "invFlightlServiceToMergeResults"

instance mergeResultsToReply memberOf Link
  hasName hasValue "mergeResultsToReply"

instance recTravelBookingRequest memberOf Receive
  hasName hasValue "recTravelBookingRequest"
  hasCreateInstance hasValue true
  hasVariable hasValue varTravelBookingRequest
  hasPartnerLink hasValue travelBookingPL
  hasOperation hasValue "bookFlight"
  isSource hasValue receiveToSplitRequest

instance assSplitRequest memberOf Assign
  hasName hasValue "assSplitRequest"
  hasAssignOperation hasValue {copyHotelRequest, copyFlightRequest}
  isTarget hasValue receiveToSplitRequest
  isSource hasValue {splitRequestToInvHotelService, splitRequestToInvFlightService}

instance copyHotelRequest memberOf Copy
  hasFromSpecification hasValue fromTravelRequestHotel
  hasToSpecification hasValue toHotelRequest

instance fromTravelRequestHotel memberOf CopyVariablePart
  hasVariable hasValue varTravelBookingRequest
  hasPart hasValue "http://ip−super.org/processes/TravelAgency.wsdl#wsdl11.messagePart(
        travelBookingRequestMessage/Hotel)"

instance toHotelRequest memberOf CopyVariablePart
  hasVariable hasValue varHotelBookingRequest
  hasPart hasValue "http://ip−super.org/processes/HotelService.wsdl#wsdl11.messagePart(
        hotelBookingRequestMessage/Hotel)"

instance copyFlightRequest memberOf Copy
  hasFromSpecification hasValue fromTravelRequestFlight
  hasToSpecification hasValue toHotelRequest

instance fromTravelRequestFlight memberOf CopyVariablePart
  hasVariable hasValue varTravelBookingRequest
  hasPart hasValue "http://ip−super.org/processes/TravelAgency.wsdl#wsdl11.messagePart(
        travelBookingRequestMessage/Flight)"

instance toHotelRequest memberOf CopyVariablePart
  hasVariable hasValue varFlightBookingRequest
  hasPart hasValue "http://ip−super.org/processes/FlightService.wsdl#wsdl11.messagePart(
        flightBookingRequestMessage/Flight)"

instance invHotelService memberOf Invoke
```

```
  hasName hasValue "invHotelService"
  hasVariable hasValue varHotelBookingRequest
  hasPartnerLink hasValue hotelBookingPL
  hasOperation hasValue "bookHotel"
  isTarget hasValue splitRequestToInvHotelService
  isSource hasValue invHotelServiceToMergeResults

instance invFlightService memberOf Invoke
  hasName hasValue "invFlightService"
  hasVariable hasValue varFlightBookingRequest
  hasPartnerLink hasValue flightBookingPL
  hasOperation hasValue "bookFlight"
  isTarget hasValue splitRequestToInvFlightService
  isSource hasValue invFlightServiceToMergeResults

instance assMergeResults memberOf Assign
  hasName hasValue "assMergeResults"
  hasAssignOperation hasValue {copyHotelResponse, copyFlightResponse}
  isTarget hasValue {invHotelServiceToMergeResults, invFlightlServiceToMergeResults}
  isSource hasValue mergeResultsToReply

instance copyHotelResponse memberOf Copy
  hasFromSpecification hasValue fromHotelResponse
  hasToSpecification hasValue toTravelResponseHotel

instance fromHotelResponse memberOf CopyVariablePart
  hasVariable hasValue varHotelBookingResponse
  hasPart hasValue "http://ip-super.org/processes/HotelService.wsdl#wsdl11.messagePart(
        hotelBookingResponseMessage/Hotel)"

instance toTravelResponseHotel memberOf CopyVariablePart
  hasVariable hasValue varTravelBookingResponse
  hasPart hasValue "http://ip-super.org/processes/TravelAgency.wsdl#wsdl11.messagePart(
        travelBookingResponseMessage/Hotel)"

instance copyFlightResponse memberOf Copy
  hasFromSpecification hasValue fromFlightResponse
  hasToSpecification hasValue toTravelResponseFlight

instance fromFlightResponse memberOf CopyVariablePart
  hasVariable hasValue varFlightBookingResponse
  hasPart hasValue "http://ip-super.org/processes/FlightService.wsdl#wsdl11.messagePart(
        flightBookingResponseMessage/Flight)"

instance toTravelResponseFlight memberOf CopyVariablePart
  hasVariable hasValue varTravelBookingResponse
  hasPart hasValue "http://ip-super.org/processes/TravelAgency.wsdl#wsdl11.messagePart(
        travelBookingResponseMessage/Flight)"

instance repTravelBookingResponse memberOf Reply
  hasName hasValue "repTravelBookingResponse"
  hasVariable hasValue varTravelBookingResponse
  hasPartnerLink hasValue travelBookingPL
  hasOperation hasValue "bookFlight"
  isTarget hasValue mergeResultsToReply
```

Listing 7. Virtual Travel Agency Process

Semantic EPC: Enhancing Process Modeling Using Ontology Languages

Oliver Thomas, Michael Fellmann

Institute for Information Systems (IWi)
at the German Research Center for Artificial Intelligence (DFKI),
Saarbruecken (Germany)
{oliver.thomas|michael.fellmann}@iwi.dfki.de
http://iwi.dfki.de

Abstract. In this article we describe a semantic extension of event-driven process chains, with which it is possible to specify the semantics of individual model elements as it is indicated by their label in natural language using concepts of a formal ontology. To do so, a multi-level approach was developed, which comprises an ontology level, a metadata level, as well as a model level. With the approach presented here, ambiguity that is introduced by the use of natural language in semi-formal models can be removed. Moreover, new possibilities of reasoning over business process models are introduced which improve the analysis, search and validation of business processes.[1]

Keywords: Process Modeling, Modeling Languages, Event-Driven Process Chain, Semantic Web, Enterprise Ontologies, Ontology Languages

1. Introduction

A multitude of modeling languages for the representation of processes have been developed since the first large data processing applications [4]. Examples are the Petri net [26], the event-driven process chain [23], the UML activity diagram [6] or the Business Process Modeling Notation (BPMN) [7]. The models described by these modeling languages serve the communication between employees in an organization with specialist knowledge and those, with methodical or technical knowledge such as for example, consultants or software engineers [29]. One tries to avoid the problem of fuzziness in natural language and the many problems in the inherent impracticability of mathematical formulations through semi-formal, graphic forms of representation in modeling languages. These are based closely on specialized business terms, exact enough, however that the models can serve as a starting point for the implementation of computer-supported information systems.

[1] An extended version of this article will appear in the Special Issue on Information Modeling and Ontologies of the International Journal of Interoperability in Business Information Systems (http://ibis-journal.net/).

Even though this is a fundamental idea for the model-driven development of information systems [14; 17], the said linkage between natural language and graphic representation forms is a main problem of semi-formal modeling languages. The identifiers of the individual elements of a business process model are added in a natural language by the modeler, irrespective of his decision for a certain modeling language. An essential part of the semantics of a process model is thus always bound to the natural language, which, with its ambiguities, allows much room for interpretation. This is not a problem as long as a model is created and read by only one person. Clearly defined semantics for each model element is however necessary, if process models from various modelers are combined, searched and translated [28] or if it is planned that the semantics in the models should be automatically validated and used for the configuration of an information system .

The problem mentioned above can be met through the linkage of the elements of a business process model with concepts from an ontology. In this article we will develop such a semantic extension for a process modeling language, which represents the semantics of the labels of process model elements with concepts of a formal ontology. This semantic extension will be carried out exemplified by the EPC. We selected the process modeling language EPC because of its popularity in modeling practice. However, our approach is principally transferable to other semi-formal modeling languages, such as for example the UML activity diagram or BPMN.

2. Related Work

The idea of using ontologies in the area of business process management is not new. For example, Wand and Weber have used ontologies to describe and evaluate certain aspects of modeling languages [31; 32].

The core area of related work can be found at the intersection of business process management and semantic web, which was currently discussed in the workshop "Semantics for Business Process Management" at the ESWC 2006 [8]. In addition to application possibilities in industry, the usability of ontologies in bridging of semantic differences for administrative processes was exemplified [20]. However, there was no contribution showing a framework for the interplay of process modeling languages and ontologies.

While our approach to the annotation of business process models is, in principle, designed language-independent, there are related projects that are geared exclusively to the semantic annotation of models in a certain language. An approach to semantic annotation for Petri nets [11], a formal framework for process description [15], as well as a tool for the semi-automatic completion of models during model construction on the basis of similarity analyses exist for example [10]. A concept for the automatic synthesis and modification of models after changes to sub-processes also exists for the UML activity diagram [24].

While we focus more on business-level process models, the potential of combining process models with (semantic) web services is described in [18; 19]. This work can be seen as complementary to our approach and might be used in the future in order to provide a framework for the integration of semantic business-level and IT-level process models.

Semtalk is a tool for the linkage of EPC-models with ontologies on the basis of Microsoft VISIO [12]. However, with this tool the semantics of the EPC-model elements is bound to the properties and operations of objects (in the object-oriented meaning), which heavily limits the usability of the modeling language.

3. Research Methodology

With the approach presented here, the semantics of individual model elements will be specified using concepts from a formal ontology. The linkage of model elements with the ontology required for this will be realized using a separate metadata level. Thus, the modeling tools and data formats remain usable while the metadata can be saved in formats accessible to the direct machine processing of the semantics contained in the models.

Altogether, the connections illustrated in the framework for the semantic annotation of business process models exist between models, metadata and ontologies (cp. Fig. 1). Metadata is generated from models (arrow from "Models" to "Metadata"). This metadata contains references to the model elements of the initial model, as well as to the concepts of the ontology. Ontologies and metadata are interdependent (double-headed arrow between "Ontologies" and "Metadata"). Concepts from the ontology are used in the metadata to specify the meaning of model elements. Therefore, the ontologies used must contain the required concepts or they must be added to the ontologies in the course of the creation of the metadata.

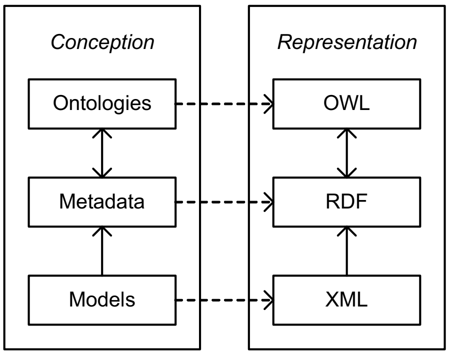

Fig. 1. Framework for the semantic annotation of business process models

The conceptual elements of the approach presented here can be assigned to representation formats for implementation purposes. These can be seen on the right side of Fig. 1 and will be introduced at a later point in time.

In the course of this article, we will first discuss ontologies for business process management. Then, in the main part of the article, we will show how ontologies and event-driven process chains can be combined to form an integrated approach to semantic business process modeling. Finally, the article closes with a discussion and an outlook.

4. Ontologies for Semantic Business Process Management

A standardization of terms for and concepts on ontologies has been the topic of research for years in the field of artificial intelligence and the semantic web. According to Gruber, an ontology is "a formal, explicit specification of a shared conceptualization" [16]. In this article, we transfer the basic idea of the semantic web which is to give information a well-defined meaning in order to make it processable both for humans and machines [9], to the field of business process management. In our approach, ontologies are not only used to clarify the semantics of individual model elements, but also to infer new facts not included in the original process model to enable advanced search and validation capabilities (see also section 5.3).

There are various languages for the explicit and formal representation of an ontology such as, for example CML, Conceptual Representation, CycL, KIF, Loom, OIL and the Web Ontology Language (OWL). OWL [1] is a standard from the World Wide Web Consortium (W3C), which resulted from the merging of DARPA and OIL. OWL will be used here as the language for representing ontologies due to its increased acceptance and, in connection with this, the support of the language through software libraries and tools. OWL is available in three variations: OWL Lite, OWL DL and OWL Full, however, the level "DL" is sufficient for the ontologies discussed in this article.

It is unnecessary to develop completely new ontologies for semantic business process management. First, one should leverage existing ontologies. In the area of enterprise and process modeling, relevant ontologies include the Enterprise Ontology [30], TOVE [TOronto Virtual Enterprise, 13] and BMO [Business Management Ontology, 22]. These ontologies provide a starting point for the coherent description of the enterprise. Second, the definitions for ontology-construction found in established technical standards and vocabularies can be reused as valuable assets. These are, for example, in the business processes field ebXML and RosettaNet, for business transactions EDIFACT and OpenTrans, for business documents UBL and xCBL, for the classification of products and services UNSPSC, eCl@ss, cXML and ISIC – to name but a few. In addition to these enterprise-spanning standards, ontologies can, third, also be obtained from the company-specific conceptualization of a domain. For this, ontologies can also be derived from entity relationship models common in the environment of relational databases and ERP-systems using the Ontology Definition Metamodel (ODM) [3] proposed by the OMG.

In the following, we will show a simple example of an ontology and illustrate it with a graphic representation (cp. Fig. 2). Properties symbolized by arrows signify object properties (ObjectProperties) in OWL, which correlate the instances of classes to one another. Inheritance relations refer to the language construct `rdfs:subClassOf` used in RDF and OWL.

The ontology framework exemplarily contains classes for organizational units, tasks, events, services and rules as relevant elements of an enterprise description. These classes can be specialized arbitrarily. In our example, the classes `Event` and `Service` were further specialized (cp. Fig. 2). In addition to classes, the example ontology contains instances, which symbolize a member of a class. The properties `partOf` and `uses` are defined to be transitive, so that additional facts can be inferred by querying the ontology with query languages. In the course of this article, our example ontology will be used to specify the model element-specific semantics of the elements of an EPC-model.

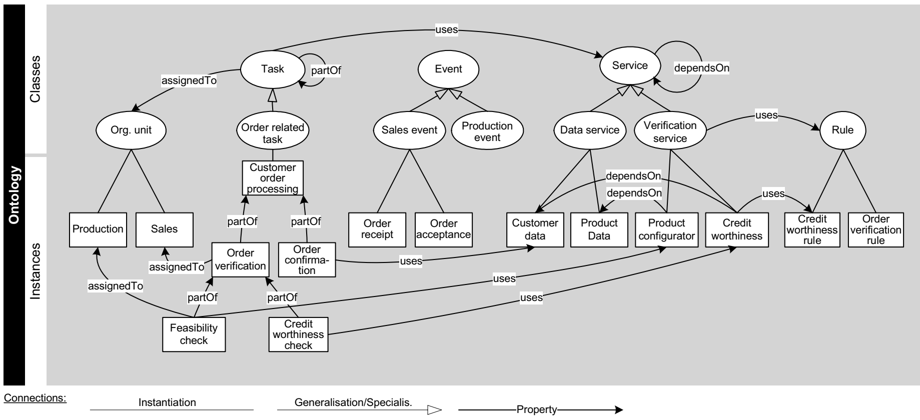

Fig. 2. Framework for an enterprise ontology

5. Semantic Event-Driven Process Chains

5.1. The Modeling Language EPC

The event-driven process chain is a modeling language for the representation of business processes common in research and practice. It was developed at the Institute for Information Systems at the Saarland University in Saarbruecken, in cooperation with the SAP, Inc. [23]. An EPC-model is a directed and connected graph, whose nodes are events, functions and logical connectors. Fig. 3 shows an example EPC-model, which describes the process for customer order processing.

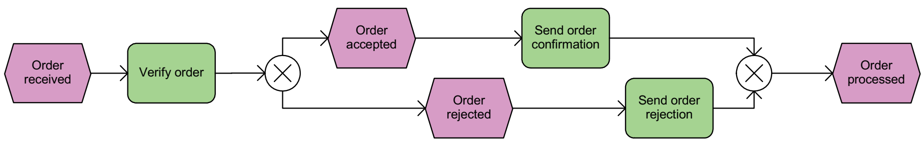

Fig. 3. EPC-model for customer order processing

Events are the passive elements in the EPC and are represented by hexagons. Functions, represented by rounded rectangles, are the active elements in the EPC. The term "function" is equivalent to the term "task" in the EPC [23]. While functions represent time-consuming happenings, events occur at a certain point in time. In literature, the respective object and an infinitive verb are suggested as a naming convention for functions, whereas for events, the object that experiences the change, as well as a verb in perfect tense, which states the type of change are suggested [27]. Events trigger functions and are their result. Control flow edges represent the relationships between functions and events. Conjunctive "$\wedge$", adjunctive "$\vee$" and disjunctive "$\otimes$" logical connectors are introduced to express that functions are started by one or more events resp. that a function can create one or more events as a result (cp. Fig. 3). They are referred to as AND-, OR- resp. XOR-connectors.

5.2. Ontology-based Representation of the EPC

To specify the semantics of EPC-model elements through relations to ontology concepts, the EPC first must be represented within the ontology. In regard to the representation of the EPC in the ontology, one can differentiate between a representation of EPC-language constructs and a representation of EPC-model elements. EPC-language constructs such as "function" or "event", as well as the control flow are created in the ontology as classes and properties. Subsequently, the EPC-model elements can be represented through the instantiation of these classes and properties in the ontology. Fig. 4 shows this by means of a simple process fragment.

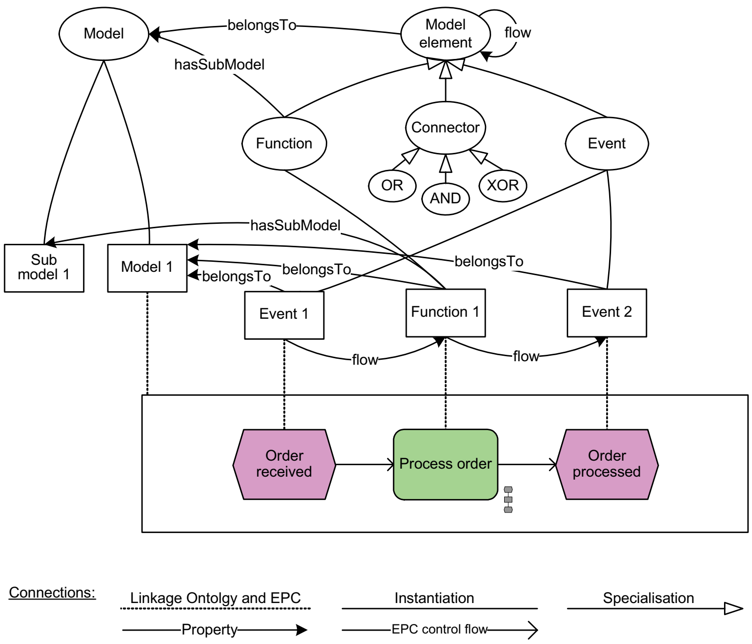

Fig. 4. Representation of the EPC in the ontology

5.3. The Linkage between EPC-Model Elements and Ontology Instances

The linkage of EPC-model elements with ontology instances can also be referred to as a process of semantic annotation. The EPC-model elements already represented in the ontology (cp. preceding section) are thereby put in relation to further instances of the ontology. Fig. 5 shows this linkage based on the example process of Fig. 3 and the example ontology represented in Fig. 2. The linkage of the ontology and EPC-model element instances is accomplished by the usage of properties; these are represented in Fig. 5 as `semType`-properties. Just as the name indicates, these properties specify the semantics of an EPC-model element through a relation to an ontology instance with formal semantics defined by the ontology.

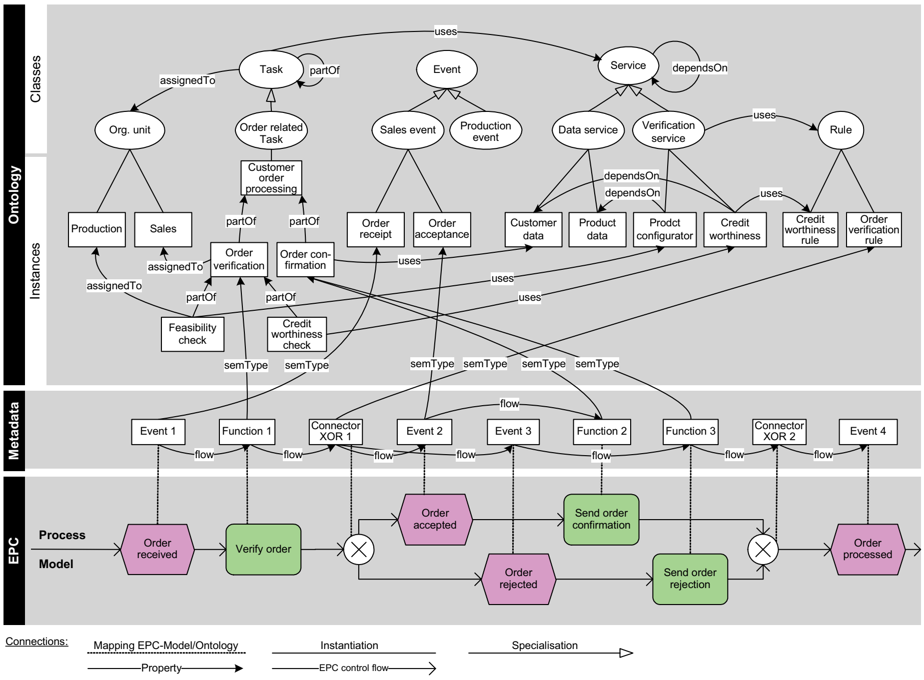

Fig. 5. Semantically annotated process model "customer order processing"

In addition to the decoupling of the semantics of an individual model element from its natural language label, the context of a model element is specified more accurately through the linkage of an ontology instance to the model element. This happens via relations, which exist between the ontology instance representing the EPC-model element and further instances of the ontology. In principle, such a specification of relations to further concepts, such as organizational units or resources, was already suggested with the extended EPC [27] and other approaches to multi-perspective modeling. In contrast to these approaches, the concept presented here uses a flexible, graph-based data model, which allows machine-processable semantics that can be extended by integrating rules.

By means of the graph-based data model provided by the Resource Description Framework (RDF) [2] and OWL, a business process is represented in the semantic metadata as an directed graph with nodes and edges. Consequently, one can traverse the graph jumping from one node to the next via properties using simple patterns, also referred to as graph pattern matching. An example for such a query is the question in the example in Fig. 5, as to whether an EPC-function exists, connected via a property `semType` to a `Task`, whose parts are connected via a property `uses` with instances of the class `Service`, which in turn are connected via a property `uses` with an instance of the class `Rule`. With SPARQL [5], which is recommended by the W3C, we already have a query language for carrying out such queries.

Moreover, new facts that are not explicitly created in the process model by the modeler can be inferred during the execution of the query. In the example in Fig. 5, one can conclude through the transitive definition of the property `partOf`, that the `feasibility check` is a part of `customer order processing`. Rule languages allow a significant extension of the machine-processable semantics. Rules can be embedded in the OWL-ontology using SWRL (Semantic Web Rule Language) [21]. SWRL rules can be expressed using the syntax of OWL, therefore allowing a tight integration of ontologies and rules. An example for a simple rule is the uncle-rule, which implies an uncle-relation through the composition of parent and brother-relations:

$$\text{parent}(?x,?y) \wedge \text{brother}(?y,?z) \;\Rightarrow\; \text{uncle}(?x,?z)$$

Transferred to business process modeling, such rules allow, as integrity rules, an advanced semantic validation. Thus, for example, the policy can be formulated that all business process related to “order processing” must contain a function “customer confirmation”. In addition, new facts can be won in the form of derivation rules during runtime. Thus, for example, we can conclude that a process, which contains a function that requires semi-finished products, reduces stock.

5.4. RDF-Representation of the Semantic EPC

In technical terms, the linkage of EPC-model elements is realized by adding attributes to the XML-representation of an EPC-model. These attributes identify the ontology instance which semantically specifies the relevant process model element. Fig. 6 illustrates this graphically, as well as with the corresponding XML-vocabularies EPML (Event-Driven Process Markup Language) for the EPC-representation [25], RDF for a semantic representation of the EPC – referred to as sEPC – and OWL for the representation of ontology classes and instances.

As we can see in Fig. 6, a linkage of the EPC-model element and ontology instance occurs over an intermediate step in the form of metadata. This metadata references both the ontology instance and the process model element, which is indicated by the dashed line connecting `checkOrder` in the process model, in the metadata and in the ontology (cp. also Fig. 5). In addition, the natural language labels of the EPC-model elements are used as names in the metadata in the field `rdfs:label` (cp. Fig. 6), indicated by another dashed line going from `name` in the EPML-data to `rdfs:label` in the RDF-data.

Seen from a conceptual point of view, the expressiveness of RDF is sufficient for the metadata, because language constructs from OWL are not used. Seen from a technical view however, then OWL DL is necessary, because the ontology instances used for the annotation must be imported into the metadata for querying and reasoning purposes.

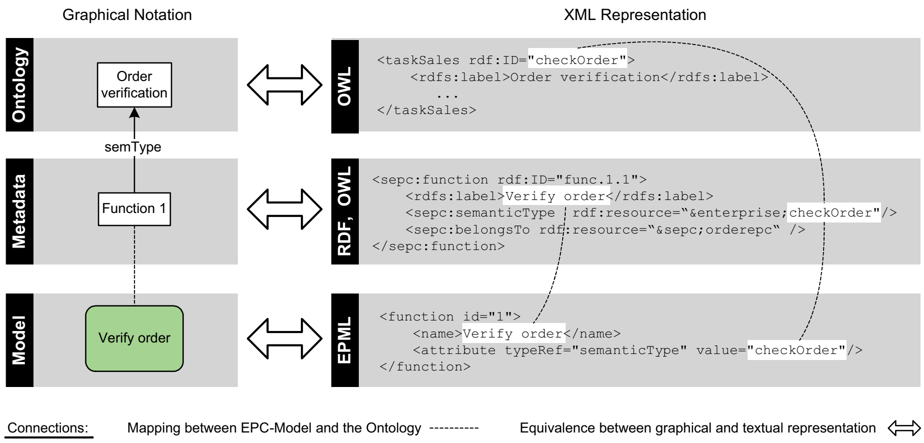

Fig. 6. Linkage of EPC-models with ontologies (representation)

After the linkage of the EPC-model with the ontology instances, a complete transformation of the EPC into an sEPC can take place on the basis of the representation formats. The sEPC consists of the XML-representation of the metadata shown exemplarily in Fig. 6. The transformation is shown in Fig. 7.

Fig. 7. Transformation from EPML to RDF

6. Conclusion and Outlook

When selecting a modeling language for the representation of business processes one must balance between formal precision and pragmatic manageability. Modeling languages with formal semantics are suited for machine processing. The interpretation of real-world interrelations can however, become very complex. With our approach, the gap between formal and semi-formal languages can be closed by linking model elements from semi-formal languages with concepts from formal ontologies and thus, receiving a formal semantic. The advantages of this transformation of process models into semantic process models using OWL are:

- *Process knowledge:* On the one hand, the understanding of business processes is increased through the linkage of model elements with the concepts of an ontology, because clearly defined terms are used and on the other, the elements of a business process are thus embedded in a certain context. This context can contain further specialized and technical information, which makes semantically annotated process models suitable as a starting point for process-oriented knowledge management.
- *Process representation:* The effort of "internationalizing" process models is reduced, because identifiers can be stored in the ontology in several languages and are thus, made usable for the automated translation of the labels of the model elements.
- *Process search:* Queries to process models can take place on the semantic level. By using inference mechanisms and rule languages, new facts not explicitly contained in the process models can be inferred at query time.
- *Process validation:* In addition to the syntactic rules defined by the meta-model of a process modeling language such as the EPC, the validation of process models can also occur on a semantic level by the usage of a rule base, which is stored in the ontology. Semantically incorrect business process models can thus be identified before process execution and policies can be enforced on all of the business processes consistently.
- *Process execution:* Process execution is simplified because the ontology acts as the central repository of a hybrid, i.e. a conceptual, as well as technical description of the elements of a business process. Best practices in the transfer of conceptual processes in IT-systems can thus be centrally stored in the ontology, free of redundancies and reusable by means of semantically annotated process models.

The need for further research with reference to the semantic annotation of process models exists regarding IT-support for the approach presented, in particular for the IT-based realization of the annotation. Interesting is also the question as to how to deal with dynamics, i.e. changes in the ontologies used for annotation, as well as the connection of the approach to semantic web services or web services repositories.

7. References

[1] Smith, M. K.; Welty, C.; McGuiness, D. L. (eds.): *OWL Web Ontology Language Guide : W3C Recommendation 10 February 2004.* W3C, 2004

[2] W3C (ed.): *Resource Description Framework (RDF)*. URL http://www.w3.org/RDF/ [20.01.2007]

[3] OMG (ed.): *Ontology Definition Metamodel : Third Revised Submission to OMG/ RFP ad/2003–03–40.* Needham : OMG, 2005

[4] Dumas, M.; van der Aalst, W. M. P.; ter Hofstede, A. H. M. (eds.): *Process-aware Information Systems : Bridging People and Software through Process Technology.* Hoboken, New Jersey : Wiley, 2005

[5] Prud‘hommeaux, E.; Seaborne, A. (eds.): *SPARQL Query Language for RDF : W3C Working Draft 23 November 2005.* W3C, 2005

[6] Object Management Group (ed.): *Unified Modeling Language: Superstructure, version 2.0, formal/05–07–04.* Needham : Object Management Group, 2005

[7] Object Management Group (ed.): *Business Process Modeling Notation Specification : Final Adopted Specification dtc/06–02–01.* Needham : Object Management Group, 2006

[8] Hinkelmann, K.; Karagiannis, D.; Stojanovic, N.; Wagner, G. (eds.): *Proceeding of the Workshop on Semantics for Business Process Management at the 3rd European Semantic Web Conference 2006, Budva, Montenegro, June 2006*

[9] Berners-Lee, T.; Hendler, J.; Lassila, O.: *The Semantic Web.* Scientific American, 2001

[10] Betz, S.; Klink, S.; Koschmider, A.; Oberweis, A.: Automatic User Support for Business Process Modeling. In: *Proceeding of the Workshop on Semantics for Business Process Management at the 3rd European Semantic Web Conference 2006, Budva, Montenegro, June 2006*, pp. 1–12

[11] Brockmans, S.; Ehrig, M.; Koschmider, A.; Oberweis, A.; Studer, R.: Semantic Alignment of Business Processes. In: Manolopoulos, Y.; Filipe, J.; Constantopoulos, P.; Cordeiro, J. (eds.): *Proceedings of the Eighth International Conference on Enterprise Information Systems (ICEIS 2006).* Paphos, Cyprus : INSTICC Press, 2006, pp. 191–196

[12] Fillies, C.; Weichhardt, F.: *On Ontology-based Event-driven Process Chains*. URL http://www.semtalk.com//pub/semtalkepk.pdf [25.11.2005]

[13] Fox, M. S.: The TOVE Project: A Common-sense Model of the Enterprise. In: Belli, F.; Radermacher, F. J. (eds.): *Industrial and Engineering Applications of Artificial Intelligence and Expert Systems, 5th International Conference, IEA/AIE – 92, Paderborn, Germany, June 9–12, 1992, Proceedings.* London : Springer, 1992 (LNCS), pp. 25–34

[14] Frankel, D. S.: *Model driven architecture : applying MDA to enterprise computing.* Indianapolis, IN : Wiley, 2003

[15] Greco, G.; Guzzo, A.; Pontieri, L.; Saccá, D.: An Ontology-Driven Process Modeling Framework. In: Galindo, F.; Takizawa, M.; Traunmüller, R. (eds.): *Database and Expert Systems Applications: 15th International Conference, DEXA 2004, Zaragoza, Spain, August 30-September 3, 2004. Proceedings.* Berlin : Springer, 2004 (LNCS), pp. 13–23

[16] Gruber, T. R.: Toward principles for the design of ontologies used for knowledge sharing? In: *International Journal of Human-Computer Studies* 43 (1995), No. 5–6, pp. 907–928

[17] Hailpern, B.; Tarr, P.: Model-driven development: The good, the bad, and the ugly. In: *IBM Systems Journal* 45 (2006), No. 3, pp. 451–461

[18] 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. In: *Proceedings of the IEEE ICEBE 2005, October 18–20, Beijing, China.* Bejing, China, 2005, pp. 535–540

[19] Hepp, M.; Roman, D.: An Ontology Framework for Semantic Business Process Management. In: Oberweis, A.; Weinhardt, C.; Gimpel, H.; Koschmider, A.; Pankratius, V.;

Schnizler, B. (eds.): *eOrganisation: Service-, Prozess-, Market-Engineering : 8. Internationale Tagung Wirtschaftsinformatik ; Karlsruhe, 28. Februar – 2. März 2007 ; Band 1.* Karlsruhe : Universitätsverlag, 2007, pp. 423–440

[20] Herborn, T.; Wimmer, M. A.: Process Ontologies Facilitating Interoperability in eGovernment A Methodological Framework. In: Hinkelmann, K.; Karagiannis, D.; Stojanovic, N.; Wagner, G. (eds.): *Proceeding of the Workshop on Semantics for Business Process Management at the 3rd European Semantic Web Conference 2006, Budva, Montenegro, June 2006*, pp. 76–89

[21] Horrocks, I.; Patel-Schneider, P. F.; Boley, H.; Tabet, S.; Grosof, B.; Dean, M.: *SWRL: A Semantic Web Rule Language : Combining OWL and RuleML ; W3C Member Submission 21 May 2004.* W3C, 2004

[22] Jenz, D. E.: *Strategic White Paper: Ontology-Based Business Process Management ; The Vision Statement.* Erlensee : Jenz & Partner GmbH, 2003

[23] Keller, G.; Nüttgens, M.; Scheer, A.-W.: Semantische Prozeßmodellierung auf der Grundlage "Ereignisgesteuerter Prozeßketten (EPK)". In: Scheer, A.-W. (ed.): *Veröffentlichungen des Instituts für Wirtschaftsinformatik*, No. 89, Saarbrücken : Universität des Saarlandes, 1992 (in German)

[24] Lautenbacher, F.; Bauer, B.: Semantic Reference- and Business Process Modeling enables an Automatic Synthesis. In: Hinkelmann, K.; Karagiannis, D.; Stojanovic, N.; Wagner, G. (eds.): *Proceeding of the Workshop on Semantics for Business Process Management at the 3rd European Semantic Web Conference 2006, Budva, Montenegro, June 2006*, pp. 89–100

[25] Mendling, J.; Nüttgens, M.: *EPC Markup Language (EPML) – An XML-Based Interchange Format for Event-Driven Process Chains (EPC) : Technical Report JM–2005–03–10.* Vienna University of Economics and Business Administration, 2005

[26] Petri, C. A.: *Kommunikation mit Automaten.* Bonn : Mathematisches Institut der Universität Bonn, 1962 (in German)

[27] Scheer, A.-W.; Thomas, O.; Adam, O.: Process Modeling Using Event-driven Process Chains. In: Dumas, M.; van der Aalst, W. M. P.; ter Hofstede, A. H. M. (eds.): *Process-aware Information Systems : Bridging People and Software through Process Technology.* Hoboken, New Jersey : Wiley, 2005, pp. 119–145

[28] Thomas, O.: Joint Reference Modeling: Collaboration Support through Version Management. In: Sprague, R. H. (ed.): *Proceedings of the 40th Annual Hawaii International Conference on System Sciences : 3–6 January 2007, Big Island, Hawaii ; Abstracts and CD-ROM of Full Papers.* Los Alamitos, CA : IEEE Computer Society Press, 2007

[29] Thomas, O.; Scheer, A.-W.: Tool Support for the Collaborative Design of Reference Models – A Business Engineering Perspective. In: Sprague, R. H. (ed.): *Proceedings of the 39th Annual Hawaii International Conference on System Sciences : 4–7 January 2006, Kauai, Hawaii ; Abstracts and CD-ROM of Full Papers.* Los Alamitos, CA : IEEE Computer Society Press, 2006

[30] Uschold, M.; King, M.; Moralee, S.; Zorgios, Y.: The Enterprise Ontology. In: *The Knowledge Engineering Review* 13 (1998), No. 1, pp. 31–89. – Special Issue on Putting Ontologies to Use

[31] Wand, Y.; Weber, R.: Research Commentary: Information Systems and Conceptual Modeling – A Research Agenda. In: *Information Systems Research* 13 (2002), No. 4, pp. 363–376

[32] Weber, R.: *Ontological foundations of information systems.* Melbourne : Coopers & Lybrand and the Accounting Association of Australia and New Zealand, 1997

Towards an Ontology for Process Monitoring and Mining

Carlos Pedrinaci and John Domingue

Knowledge Media Institute, The Open University, Milton Keynes, UK.
(c.pedrinaci,j.b.domingue)@open.ac.uk

Abstract. Business Process Analysis (BPA) aims at monitoring, diagnosing, simulating and mining enacted processes in order to support the analysis and enhancement of process models. An effective BPA solution must provide the means for analysing existing e-businesses at three levels of abstraction: the Business Level, the Process Level and the IT Level. BPA requires semantic information that spans these layers of abstraction and which should be easily retrieved from audit trails. To cater for this, we describe the Process Mining Ontology and the Events Ontology which aim to support the analysis of enacted processes at different levels of abstraction spanning from fine grain technical details to coarse grain aspects at the Business Level.

1 Introduction

Business Process Management (BPM) intends to support "business processes using methods, techniques, and software to design, enact, control, and analyse operational processes involving humans, organisations, applications, documents and other sources of information" [16]. As opposed to so-called Workflow Management Systems (WFMS), BPM acknowledges and aims to support the complete life-cycle of business processes which undoubtedly involves post-execution analysis and reengineering of process models. However, by doing so BPM has made more evident the existing difficulties for obtaining automated solutions from high-level business models, and for analysing the execution of processes from both a technical and a business perspective.

The fundamental problem is that moving between the Business Level and the IT Level is hardly automated. In fact, reusing the words from [2], business modelling is not process modelling. Deriving an IT implementation from a business model is particularly challenging and requires an important and ephemeral human effort which is expensive and prone to errors. Conversely analysing automated processes from a business perspective, e.g., calculating the economical impact of a process or the performance of departments within an organisation, is again an expensive and difficult procedure which typically requires a human in the loop.

In this paper we shall focus on the transition from the IT perspective into the Business Level. First we introduce Semantic Business Process Management and present our approach to overcoming BPM limitations. Next, we focus on the mining and monitoring of processes. In particular we present initial work on the Process Mining Ontology (PMO) which aims to capture events taking place during the life-cycle of business and IT processes and combine it with additional mining information in order to support the analysis of enacted processes at different levels of abstraction spanning from fine grain technical details to coarse grain aspects at the Business Level. Finally, we summarise and identify future research that will be carried in this context.

2 Semantic Business Process Management

So far BPM has focussed mainly on supporting the graphical definition of business processes and on the derivation of skeletal executable definitions that could automate them. From the modelling perspective, notable examples are Event-driven Process Chains (EPC) [7] and the Business Process Modelling Notation (BPMN) [13]. On the technical side, the so-called Service-Oriented Architecture and related technologies such as Web Services, WS-BPEL [11] or Message-Oriented Middleware are perhaps the main enabling technologies [6].

Current approaches to BPM suffer from a lack of automation that would support a smooth transition between the business world and the IT world [5]. On the one hand current technologies only support the derivation of partial definitions of executable processes and still require an important human effort in order to obtain robust deployable solutions. On the other hand, once deployed these automated processes need to be continuously monitored, analysed, enhanced and adapted to meet evolving (business or technical) requirements and to accommodate ever-changing (business or technical) environments.

In [5] the authors argue that the difficulties for automating the transition between both worlds is due to a lack of machine processable semantics. Often business modelling is in fact approached as process modelling [2, 3], and process modelling mainly focusses on the graphical representation of processes using modelling languages, e.g., BPMN, which cannot capture domain specific semantics. As a result, processes definitions do not provide machine processable semantics that could support business practitioners in the analysis and reengineering of processes, and executable processes definitions, e.g., WS-BPEL, are bound to inflexible syntactic definitions which pose important technical difficulties.

Semantic Business Process Management that is, the combination of Semantic Web and Semantic Web Services technologies with BPM, has been proposed as a solution for overcoming these problems [5]. SBPM aims at accessing the process space of an enterprise at the Knowledge Level so as to support reasoning about business processes, process composition, process execution, etc. SBPM builds upon the use of ontologies as a core component providing the required semantic information and enhances the composition, mediation and discovery of Web Services by applying Semantic Web Services techniques.

2.1 The SUPER Approach

Major efforts are currently devoted to pursuing the SBPM vision in the context of the European project SUPER which stands for Semantic Utilised for Process Management within and between Enterprises[1]. The fundamental approach is to represent both the business perspective and the systems perspective of enterprises using a set of ontologies, and to use machine reasoning for carrying out or supporting the translation tasks between the two worlds. An initial version of a comprehensive framework conceptualising the relevant aspects for the automation of Business Process Management tasks has been devised, see Figure 1.

The stack of ontologies builds upon the use of WSMO [14] as the core Semantic Web Services conceptualisation and WSML [1] as the representation language supporting the specification of Ontologies, Goals, Web Services and Mediators. The integration between the different conceptualisations is provided by the Upper-Level Process Ontology which captures general concepts such as Process, Activity, Actor or Role which are extensively reused across the ontologies. In order to enhance the overall coherence it is envisioned that the UPO will be refined using DOLCE [9] as its foundational ontology.

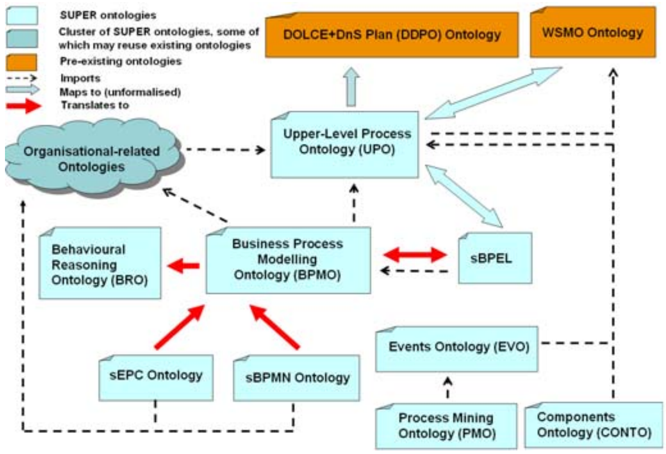

Fig. 1. SUPER Ontology Stack.

The Semantic EPC (sEPC) and Semantic BPMN (sBPMN) ontologies conceptualise EPCs [7] and BPMN [13] respectively incorporating the appropriate links to WSMO concepts. These ontologies therefore provide support for two of the main modelling notations currently used in BPM. The Business Process Modelling Ontology (BPMO) provides a common layer over both sEPC and SBPMN and links them to the rest of the ontologies from the SUPER stack. BPMO links process models to organisational information as conceptualised in the Organisational Ontologies which represent concepts like Organisation, Department, Team or Employee and the relationships between them. It is also

[1] More information at http://www.ip-super.org

linked to the Behavioural Reasoning Ontology (BRO) whose aim is to support the composition of processes by reasoning about their behaviour. Finally, BPMO enables the transformation of business processes modelled using different notations into their executable form. To support the execution of business processes, BPEL [11] has been chosen as the representation language for its extensive support and use. The Semantic BPEL (sBPEL) ontology formalises BPEL and includes additional constructs linked to WSMO so as to support the mediation between heterogeneous data or processes, or the invocation of Goals as opposed to explicitly specified Web Services.

Different transformations between these different conceptualisations have been defined, see red arrows in Figure 1. An additional transformation, although not shown in the figure, has been defined for transforming sBPEL into a serialisation format, BPEL4SWS, for executing processes on extensions of existing workflow engines. BPEL4SWS is an XML serialisation format that is mainly an extension from sBPEL with typical SWS features, e.g. including support for goals instead of predefined activities and use of mediators.

The ontology stack also identifies the Components Ontology which aims to support the conceptualisation of IT Level aspects, such as software components and systems. This ontology will be based on previous research on the semantic management of middleware [12]. Finally, because having semantics at the level of processes but not at the level of monitoring and mining defeats to an important extent the benefits that can be obtained from SBPM, the ontology stack includes two ontologies, the Process Mining Ontology and the Events Ontology which are the focus of this paper.

3 Semantic Process Monitoring and Mining

One of the distinguishing characteristics of BPM solutions with respect to traditional WFMS is commonly referred to as Business Process Analysis(BPA) [16]. In a nutshell, BPA aims at monitoring, diagnosing, simulating and mining enacted processes in order to support the analysis and enhancement of process models. The main goal pursued by BPA are on the one hand the verification or validation of the actual execution with respect to prescribed or expected processes, and on the other hand the identification, in a more or less timely manner, of potential improvements of business processes. The knowledge gained in this phase is thus employed for reengineering and fine tuning existing process definitions.

[3] identifies three main levels for the analysis of e-businesses information systems, as shown in Figure 2: the Business Level, the Process Level and the IT Level. The first level is concerned about the value exchanges between the different actors involved (e.g., companies) and is therefore of particular relevance for business practitioners. The second level considers the process point of view (e.g., BPEL level) and is usually the focus of process architects. Finally, the third level is concerned about technical details such as the decomposition of a process into Web Services. An effective BPA solution must therefore provide the means for analysing existing e-businesses at these three layers.

This layering is even more complex since there can be, and there usually are, nested layers and different perspectives that can be adopted within each of these layers. For instance the business analyst could focus on individuals, departments or the whole organisation. The process execution view might involve several (sub) processes, i.e., what appears to a process as a simple atomic task might in fact be supported by another process as is often the case for complex processes. Finally, the process execution will rely on some actual IT infrastructure which will follow some algorithm–a process in itself–which we might need to analyse.

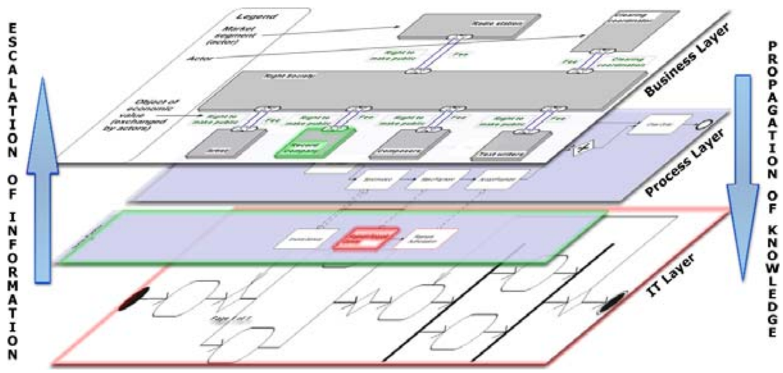

Fig. 2. Business Process Analysis layers.

Further complications come from the fact that although these layers are clearly distinguished, there exists an inherent intertwining between them. On the one hand, decisions at the Business Level have implications at the Process Level which might in turn affect the IT Level. On the other hand, the execution of some activity by some system, e.g., a Web Service, affects the Process Level and this might escalate to the Business Level. It is worth noting that this propagation between layers takes place both at design time and runtime. In fact, in some domains like telecommunications where for example quality of service is crucial, the technical details regarding the process execution are of particular relevance at the Process Level and even at the Business Level. Being able to properly correlate the data between layers at runtime can therefore be of particular importance. Automating this, as necessary for what is commonly referred to as Business Process Intelligence, requires semantic information that spans these layers of abstraction and which should be easily retrieved from audit trails.

3.1 On the Need For a Process Mining Ontology

BPA is mainly targeted at business users and process architects, although it is also concerned about the technical details since automated processes execution eventually depend on the underlying IT infrastructure. In fact, Business Activity Monitoring (BAM), one of the main areas in BPA, uses data logged by the underlying IT support in order to monitor, diagnose and mine executed processes.

In this paper we shall use *audit trail* as commonly adopted in the Workflow and Business Management communities to refer to this data.

So far, extensive work has been devoted to the definition of mechanisms for the communication of events or notifications between systems. Among this work we can mention CORBA, JMS, WS-Eventing, WS-Notification or Message-Oriented Middleware in general [6, 10]. These technologies, although not uniquely devoted to supporting monitoring, provide the necessary technical support for communicating monitoring information at runtime. What remains to be defined is an appropriate format for capturing audit trails in a way that would support the creation of fully-fledged generic BPA solutions. In fact, as often happens in IT, every specific system provides its own level of detail in heterogeneous formats making it particularly difficult to integrate the audit trails generated by diverse systems as well as it complicates the creation of general purpose BPA solutions.

Perhaps the main effort in defining a common format for storing audit trails has been undertaken in the context of the ProM framework, a pluggable environment for process mining [17]. ProM is able to apply a wide range of process mining algorithms over log data stored in MXML [18], an XML-based format that captures the necessary information for process mining. In a nutshell, MXML establishes that each audit trail should be an event happened at a specific moment during the execution of an uniquely identifiable activity. The events should specify what actually happened (e.g., start or end of an activity) and they should refer to a concrete process instance belonging to a specific process. The reader is referred to [18] for more details about the format.

MXML has proven to be suitable for capturing diverse audit trails. In particular there currently exists support for importing logs from PeopleSoft, Staffware or FLOWer to name a few [18]. Still, MXML is not all there needs to be to support SBPM. Audit trails, although generated by general purpose software, obviously concern domain specific resources. That is, work may be performed by a specific person, belonging to a specific department from a concrete office of a given company. An MXML log is only able to refer to a *label* identifying the name of this person. The actual *semantics*, i.e., who that person is, where he or she works and other related information are not available. Indeed, in many cases it may be possible to create ad-hoc solutions for retrieving this knowledge but this clearly defeats the very purpose of defining a generic format. In other words, MXML suffers from a lack of machine processable semantics as we previously identified for BPM in general. As a result navigating through the levels of abstraction required for analysing e-business solutions requires a human in the loop capable to identify the links and relations across layers.

Semantic Web technologies, in particular ontologies [4] for they are formal, sharable and extensible representations, together with the related tooling such as repositories and reasoners, offer a suitable framework on which to build upon generic BPA solutions. First, they are particularly well-suited for defining shared conceptualisations in order to support the integration of heterogeneous and inter-organisational sources of information. Second, having a formal definition they are amenable to automated reasoning, providing the flexibility required for navigating through different levels of abstraction and querying the overall body of knowledge about the business processes. Finally, ontologies are a step forward towards Business Intelligence, since they provide a natural means for defining reaction rules or applying knowledge-based techniques like Problem-Solving Methods [15], in order to intelligently adapt the behaviour of business processes.

In the next two sections we describe the Process Mining Ontology, that aims to enhance the state-of-the-art in BPA by semantically capturing audit trails and process mining details. In order to do so, we first focus on the Events Ontology which provides the core framework for capturing events generated by IT systems. Then we present the Process Mining Ontology which builds upon the former, see Figure 1, and enhances it with mining specific definitions.

3.2 Events Ontology

The Events Ontology (EVO) supports capturing events taking place during the life-cycle of both business and IT processes. It is based on MXML which is in turn based on the analysis of different types of logs in real systems [18]. Doing so ensures (i) that we capture the required information for applying several process mining algorithms;(ii) that we can import logs from some of the main existing systems, and (iii) that we can reuse and enhance ProM for mining EVO logs.

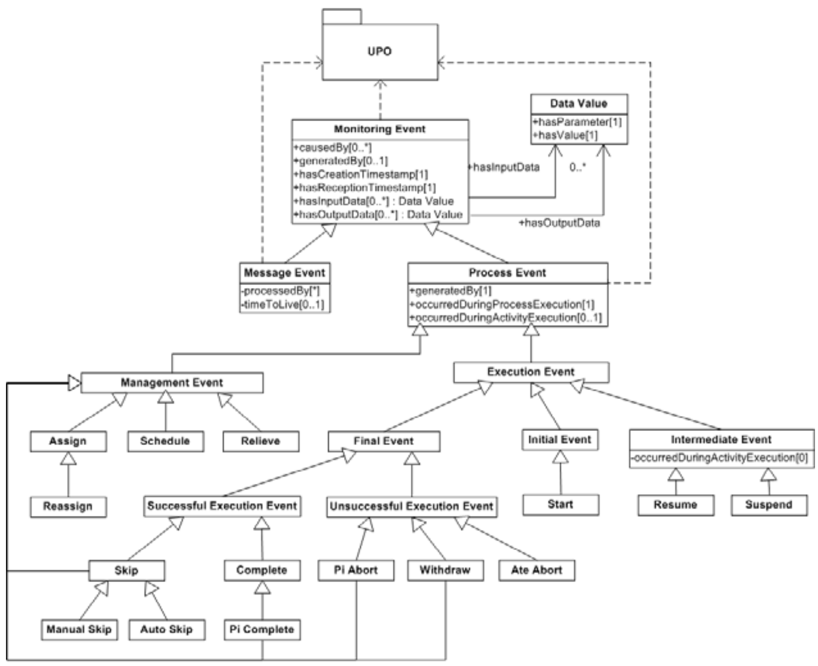

Fig. 3. Events Ontology as a UML Activity Diagram.

Figure 3 shows the EVO as a UML class diagram[2]. The main concept in EVO is *Monitoring Event*. This concept represents events generated at a specific point on time by an IT system. Monitoring Events are therefore different from "clicking a button" as usually understood in computer science. Events are generated by an *Actor* which is defined in UPO and therefore allows for reusing the rest of the ontologies defined in the SUPER stack. In particular, Actor is refined both in the Organisational Ontology, where Companies, Departments, and Individuals are defined, and in the Components Ontology which describes software components.

Time is one of the main characteristics in Event processing [8]. As a consequence, each Monitoring Event has both a *creation timestamp* based on the clock of the system where the Event was generated, and a *reception timestamp* which captures the actual moment in which the Event was received. The former accommodates pre-existing logging mechanisms and supports performance analysis at the level of every specific component since it is not influenced by external aspects such as the network latency. The latter should be introduced by the events propagation infrastructure accordingly. The reception timestamp is particularly useful for monitoring distributed systems since it supports establishing a global order among all the events without the need for clock synchronisation mechanisms and also supports detecting network malfunctions. We believe this attribute will be of most relevance given that nowadays more and more business processes are interorganizational, making the application of clock synchronisation techniques particularly challenging, if even realistic.

Events may have a set of inputs and outputs which are specified as *Data Value* instances which identify the parameter affected and the value given. Finally, an Event may have an associated *causality vector* indicating other Events which caused the generation of the former. This type of causality information is particularly relevant for processes monitoring and mining although it is rarely propagated by the runtime infrastructure excepting some Event-Based infrastructures [8]. Additionally the causality vector attribute can be used for post-execution analysis derived information if necessary.

Monitoring Events are further refined into *Message Events* and *Process Events*. Message Events accommodate Messaging Systems so that their execution can also be traced. These events are therefore placeholders for information interchanged between actors. In addition to the attributes inherited from Monitoring Event, Message Events can also capture a set of actors that have already processed the event, and can include a *Time-To-Live* parameter which basically sets an expiry date for the event.

Process Events represent events that can take place during the life-cycle of a business or IT process, thus encompassing its management and execution. Every event will affect a specific *Process* and might concern some *Activity* belonging to this Process. Process and Activity are meta-concepts defined in the UPO. This caters for distinguishing between different executions of the same Process or Activity while it allows at the same time to access their actual definitions.

[2] The complete ontology represented in WSML can be found at http://kmi.open.ac.uk/people/carlos/ontologies/PMO/evo.wsml

Specific Business and IT processes will be defined in the BPMO and the Components Ontology respectively. Thus, by abstracting away from the specificities of the process being logged, EVO provides the means for capturing audit trails at different levels of abstraction, i.e., the process and the IT level. It is worth noting that Process Events can refer to the whole Process by omitting the Activity, e.g., "a Process is suspended", or to a specific Activity within the Process, e.g. "Activity Y was correctly executed".

The EVO refines Process Events into *Management Events* and *Execution Events*. Management Events are Process Events that are not directly caused by the execution of some Process or Activity but rather generated by the action of some external actor, typically a human but possibly an automated management system. These are usually work distribution events such as *Assign*, *Re Assign*, *Schedule*, *Relieve*, and may also affect the eventual execution of the Process, e.g., *Skip*, *Pi Complete*, *Pi Abort* and *Withdraw*.

Execution Events are Events that concern the actual execution of some Process or Activity. They are further refined into *Initial Events*, *Intermediate Events* and *Final Events*. Initial Events indicate the start of the execution of a process. Intermediate Events are those Execution Events which affect the actual execution but are not Final nor Initial. This is the case for *Resume* and *Suspend*. Activities within a Process are considered atomic thus, Intermediate Events cannot refer to an Activity. Should the Activity have a complex implementation which could yield Intermediate Events, these should be logged as part of the subprocess triggered by the Activity. Last, Final Events are those that conclude the execution of a process and are categorised as *Successful* or *Unsuccessful*.

Among the remaining concepts, we next describe briefly some of the main ones captured in the ontology. The correct completion of some Process or Activity is captured by means of *Complete* events. Conversely, *Ate Abort* events indicate that the execution of the Activity or Process was aborted. The reason for the execution being aborted originates from the execution itself (e.g. software exception, unexpected result). Both *Pi Complete* and *Pi Abort* are the managerial counterpart of Complete and Ate Abort, i.e., some management action has marked the Process or Activity as *completed* or *aborted*. Furthermore, the ontology includes the typical events related to the management and scheduling of processes such as *Resume*, *Suspend*, *Assign*, *Reassign* or *Schedule*. Finally, *Skip* events indicate that the Process or the Activity has been skipped, i.e. will not be executed, and is considered as being properly executed.

3.3 Process Mining Ontology

The Process Mining Ontology formalises different aspects relevant for mining and analysing business or IT processes[3]. The PMO integrates all the diverse knowledge required for mining processes by reusing additional ontologies from the SUPER ontology stack such as the EVO, the Organisational Ontology, the

[3] The complete ontology represented in WSML can be found at http://kmi.open.ac.uk/people/carlos/ontologies/PMO/pmo.wsml

BPMO and the Components Ontology, see Figure 1. PMO depends directly on the EVO since it reuses the concepts defined therein, although they remain as separate ontologies for modularity and reusability reasons. Indirectly it integrates through the UPO, the conceptualisations captured in the Organisational Ontology, the BPMO and the Components Ontology.

The PMO, shown in Figure 4, defines containers for workflows, processes and message events audit trails so as to provide the appropriate perspective for workflow, process or Messaging Systems monitoring and mining respectively. These containers are the *Message Events Log*, the *Process Execution Log* and the *Workflow Log* concepts. It is worth noting that although it would be possible to obtain the containers by means of queries, an explicit definition allows for attaching any analysis result obtained to them for future reuse. The PMO includes a default instance both for capturing Message Event Logs and Workflow Logs, and includes an axiom for the automated creation of instances of Process Execution Log whenever events concerning a new process are received. Further axioms ensure the events are automatically added to the corresponding containers.

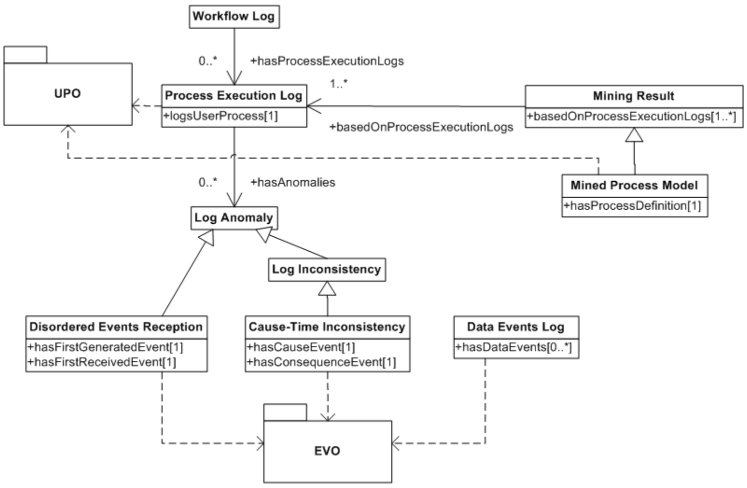

Fig. 4. Process Mining Ontology as a UML Activity Diagram.

The current version of the PMO includes an initial set of mining specific axioms in an attempt to enhance current mining support with automated detection of some anomalies. Currently, the ontology supports detecting the disordered reception of events and inconsistent cause-time relationships between events. The former anomaly, which is typically due to network problems, is detected based on the use of the two timestamps captured in Monitoring Events. Basically, an axiom checks wether the events generated by some actor are received in the same order. The detection of inconsistent cause-time relationships is particularly relevant for event analysis and is based on the causality vector captured in Monitoring Events. The relationship is considered inconsistent when the cause for an event is received after the consequent event. Again, the most likely situation where this could happen is due to network communication problems between the system being monitored and the monitoring infrastructure.

It is important to note that instead of including these axioms as constraints that would basically avoid the existence of such anomalies, we define them as logic programming rules that detect and capture the anomalies as part of the conceptualisation, see Figure 4. In fact, it is well known that often systems logs present anomalies but eliminating these would also get rid of important information. For instance, the mere fact of knowing that the reception of events is disordered is relevant information from the IT perspective. Thus, detecting and capturing these avoids using anomalous data for some analysis processes while it provides and maintains valuable information for others.

Finally, the current version of the PMO defines a placeholder for capturing mining results. In particular we currently identify the super concept *Mining Result* which represents the results obtained from applying mining algorithms over the set of logs identified by the attribute *Based On Process Execution Logs*. *Mined Process Model* is the only Mining Result defined so far although we foresee concepts like the *Mined Organisational Model* or performance related results. Future research will be devoted to defining these concepts and to refining the anomalies conceptualisation.

4 Conclusions

In this paper we have introduced the lack of automation existing in state-of-the-art BPM solutions. This drawback has been attributed to a lack of machine processable semantics and we have presented Semantic Business Process Management as approached in the SUPER project as a solution. We have focussed in particular in the challenges for the monitoring and mining of processes from a semantics perspective and we have argued for the need of ontologies to support these tasks. Finally we have described in depth the Events Ontology and the Process Mining Ontology that aim to support the analysis of enacted processes at different levels of abstraction spanning from fine grain technical details to coarse grain aspects at the business level.

The ontologies, although still subject of research and improvement are built upon solid bases stemming from one of the most complete general purpose mining solutions. The ontologies are part of an extensive formalisation of the BPM domain and therefore allow accessing the whole body of knowledge about processes, organisations or IT systems in order to support making queries at different levels of abstraction. Future research will be devoted to applying these ontologies for the monitoring and mining of the various use-cases of SUPER, using an extended version of ProM. Out of this experiments, we expect to be able to assess the suitability and benefits from using EVO and PMO for monitoring and mining as well as we might identify potential improvements for the ontologies during this process.

References

1. Jos de Bruijn. D16.1v0.21 the web service modeling language wsml. http://www.wsmo.org/TR/d16/d16.1/v0.21/, October 2005.
2. J. Gordijn, H. Akkermans, and H. van Vliet. Business modelling is not process modelling. In *Conceptual Modeling for E-Business and the Web, LNCS 1921 (ECOMO 2000)*, pages 40–51, Salt Lake City, USA, 2000. Springer-Verlag.
3. Jaap Gordijn and Hans Akkermans. E3-value: Design and evaluation of e-business models. *IEEE Intelligent Systems*, 16(4):11–17, 2001.
4. T. R. Gruber. A translation approach to portable ontology specifications. *Knowledge Acquisition*, 5(2):199–220, 1993.
5. Martin Hepp, Frank Leymann, John Domingue, Alexander Wahler, and Dieter Fensel. Semantic business process management: A vision towards using semantic web services for business process management. In Francis C. M. Lau, Hui Lei, Xiaofeng Meng, and Min Wang, editors, *ICEBE*, pages 535–540. IEEE Computer Society, 2005.
6. Gregor Hohpe and Bobby Woolf. *Enterprise Integration Patterns: Designing, Building, and Deploying Messaging Solutions*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2003.
7. G. Keller, M. Nüttgens, and August-Wilhelm Scheer. Semantische prozessmodellierung auf der grundlage "ereignisgesteuerter prozessketten (epk)". Arbeitsbericht Heft 89, Institut für Wirtschaftsinformatik Universität Saarbrücken, 1992.
8. David C. Luckham. *The Power of Events: An Introduction to Complex Event Processing in Distributed Enterprise Systems*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2001.
9. Claudio Masolo, Stefano Borgo, Aldo Gangemi, Nicola Guarino, Alessandro Oltramari, and Luc Schneider. Wonderweb deliverable d17. the wonderweb library of foundational ontologies and the dolce ontology, August 2002.
10. Gero Mühl, Ludger Fiege, and Peter Pietzuch. *Distributed Event-Based Systems*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006.
11. OASIS Web Services Business Process Execution Language (WSBPEL) TC. Web services business process execution language version 2.0 committee specification. http://docs.oasis-open.org/wsbpel/2.0/CS01/wsbpel-v2.0-CS01.pdf, January 2007.
12. Daniel Oberle. *Semantic Management of Middleware*, volume I of *The Semantic Web and Beyond*. Springer, New York, FEB 2006.
13. Object Management Group. Business process modeling notation specification - final adopted specification. Available at http://www.bpmn.org, February 2006.
14. Dumitru Roman, Holger Lausen, and Uwe Keller. D2v1.3. web service modeling ontology (wsmo). http://www.wsmo.org/TR/d2/v1.3/, October 2006.
15. R. Studer, R. Benjamins, and D. Fensel. Knowledge engineering: Principles and methods. *Data Knowledge Engineering*, 25(1-2):161–197, 1998.
16. Wil M. P. van der Aalst, Arthur H. M. ter Hofstede, and Mathias Weske. Business process management: A survey. In *Business Process Management*, pages 1–12, 2003.
17. B.F. van Dongen, A.K.A. de Medeiros, H.M.W. Verbeek, A.J.M.M. Weijters, and W.M.P. van der Aalst. The prom framework: A new era in process mining tool support. In *Applications and Theory of Petri Nets 2005. 26th International Conference, ICATPN 2005*, pages 444–454, Miami, USA, June 2005. Springer-Verlag.
18. Boudewijn F. van Dongen and Wil M. P. van der Aalst. A meta model for process mining data. In *EMOI-INTEROP*, 2005.

Semantically enhanced Business Process Modelling Notation

Witold Abramowicz, Agata Filipowska, Monika Kaczmarek, Tomasz Kaczmarek

Department of Information Systems, Poznań University of Economics
Al. Niepodległości 10, Poznań, Poland
{w.abramowicz, a.filipowska, m.kaczmarek, t.kaczmarek}@kie.ae.poznan.pl

Abstract. This position paper presents the semantically enhanced Business Process Modelling Notation, namely the sBPMN ontology, developed within the SUPER project. Moreover, it elaborates shortly on drawbacks of the BPMN to BPEL translation and proposes to use semantics to overcome them.

Keywords: Business Process Modelling Notation, ontologies, Business Process Management (BPM)

1 Introduction

Business Process Modelling Notation (BPMN) [1], created by the BPMI group, has emerged as a standard notation for process modelling, joining many other notations e.g. UML ADs, IDEF, ebXML and EPCs. It enjoys fast growing popularity among tool vendors.

BPMN aims at bridging the gap between business process design and process implementation. It was to allow for the automatic translation from the graphical process diagram into the BPEL process representation [4] that may be then executed using a Web services technology.

Although the goal of automatic translation is very appealing, the intention failed in practice for a number of reasons. One of them is that BPMN is a graph-oriented language and its mapping to the block-structured BPEL representation is challenging. In addition, BPMN allows designing ill-formed processes that cannot be translated directly into a set of the BPEL executable instructions [2].

Nowadays (May, 2007), the BPMN modelling is supported by more than 30 tools. Some of them allow also for the translation from BPMN diagrams to BPEL, but this functionality is neither fully automated nor supported with semantics. Creation of the sBPMN (Semantic Business Process Modelling Notation) ontology will add meaning to each of the process elements and make them machine-readable. In addition, it will also allow for reasoning on the process description. Once sBPMN is enhanced with Semantic Web services (SWS) extensions it will be also possible to automatically assign Web service (or their composition) to each task. Having Web services matched to tasks is only one step from generating BPEL process representation that may be deployed on the execution engine. This position paper elaborates shortly on the sBPMN ontology created within the SUPER project [3] funded by EC under the 6th Framework Programme.

2 Ontology development and modelling decisions

The sBPMN ontology was developed based on the latest available BPMN Specification [1]. The process of development of the sBPMN ontology was divided into three phases. During the first phase the top-down approach was taken to formalise the ontology using WSML [5]. In the second phase the consistency was tested by describing a number of BPMN diagrams using the designed ontology. The third phase introduced further improvements to the ontology on the basis of the annotated examples as well as requirements of the interested parties.

During the ontology development a number of modelling decisions had to be taken. First, the scope of the notation to be ontologised had to be identified. Then the selected concepts and their properties were modelled. At this stage it had to be decided whether specific occurrences of process model elements are to be subclasses or instances of specific concepts. A decision was to use Class to represent a type of entity, i.e. process, task, gateways. Therefore, core business process diagram elements [1] were modelled as classes having appropriate attributes defined in the BPMN Specification. Therefore, the annotation of processes with sBPMN means creating instances of its concepts, e.g. task ObtainLicense will be an instance of the Task class, and not a subclass of it. Another issue concerns the association of the BPMN elements to a specific process. To make sure that all the elements of a process model refer to it, special property was introduced (named hasProcess) for explicit or implicitly (through recursion) reference.

The sequence flow is modelled using the connection rules attached to the Source and Target properties of the SequenceFlow concepts defining which Flow Objects (e.g. Tasks, Activities, Events, Gateways) may be connected one to another (in line with the BPMN Sequence Flow Connection Rules). The message flow connection rules were implemented analogically.

3 Domain Captured

The core element of the sBPMN ontology is a Business Process Diagram presenting the process model. According to the BPMN specification four basic categories of elements are Flow Objects, Connecting Objects, Swimlanes and Artefacts. However, for the sake of clarity and compatibility to the other SUPER process ontologies [3], the Process concept had to be introduced at the same level. Therefore, the main concepts of the sBPMN ontology are as follows:

- Flow Objects - the main graphical elements defining the behaviour of a Business Process. There are three kinds of Flow Objects: Events, Activities and Gateways.
- Connecting Objects – as there are three ways of connecting the Flow Objects to each other or to other resources, BPMN utilises three types of Connecting Objects: Sequence Flow, Message Flow and Association.

- Swimlanes - utilised when grouping the primary modelling elements (see above). Two kinds of swimlanes were developed in BPMN, namely: Pools and Lanes.
- Artefacts - used to provide additional information about the process. The current set of Artefacts includes: Data Object, Group and Annotation.
- Process - used to group flow objects elements into a set of objects.

The above concepts represent only the core subset of the sBPMN ontology. Each of them has a number of subconcepts and so on. The current sBPMN ontology has 95 concepts and over 50 axioms. It is available at the SUPER project website (http://www.ip-super.org).

4 Competency Questions

The designed sBPMN ontology was verified describing a few exemplary BPMN process diagrams. After successful creation of the semantic descriptions, it was then tested against the competency questions, e.g. what are the elements of a given process, what are the sequence flow connection rules, what is the execution order of activities within the process, which objects can be a source of compensation association, how a certain type of activity can be triggered, etc. As a result the domain coverage as well as reasoning possibilities were proved.

4 Conclusions and future work

sBPMN ontology is to overcome problems with composition and execution of processes based on their models designed by business analysts. It does not enforce well-formedness of the process models in itself. Yet, additional set of axioms could be devised, which, together with semantic business process model, could lead to automatic assessment of well-formedness of the model using standard reasoning techniques. The current version of the sBPMN ontology is used to create the semantic annotation of the processes within the modelling tool proposed by the SUPER project. In the future the ontology will be further incorporated in the ontology stack developed within the project.

Acknowledgments. The work published in this article was partially supported by the SUPER project (http://www.ip-super.org) under the EC 6th Framework Programme Information Society Technologies Objective (contract no. FP6-026850).

References

1. Business Process Modeling Notation Specification. OMG Final Adopted Specification, February 6, 2006

2. Ouyang C., van der Aalst, W.M.P., Dumas, M., ter Hofstede, A.H.M., Translating BPMN to BPEL. BPM Center Report BPMcenter.org, 2006 available at http://is.tm.tue.nl/staff/wvdaalst/BPMcenter/reports.htm
3. Business Process Ontology Framework, deliverable 1.1, SUPER project, no. FP6-026850
4. Arkin, A., Askary, A., Bloch, B., Curbera, F. et al: Web Services Business Process Execution Language Version 2.0. Working Draft. WS-BPEL TC OASIS, May 2005
5. Web Service Modelling Language http://www.wsmo.org/wsml/

Semantic Business Process Repository

Zhilei Ma[1], Branimir Wetzstein[1], Darko Anicic[2], Stijn Heymans[2], Frank Leymann[1]

[1]Institute of Architecture of Application Systems (IAAS)
University of Stuttgart, Germany
{firstname.lastname}@iaas.uni-stuttgart.de
[2]Digital Enterprise Research Institute (DERI)
University of Innsbruck, Austria
{firstname.lastname}@deri.org

Abstract. Semantic Business Process Management (SBPM) utilizes semantic technologies to achieve more automation throughout the BPM lifecycle. An integral part of the SBPM infrastructure is a semantic business process repository, which is used for storage and management of business process modeling artifacts. As in SBPM business process models are based on process ontologies, the semantic business process repository has additional requirements towards support of reasoning and querying capabilities. In this paper, we first describe the functionalities the semantic business process repository has to provide. We then introduce a solution based on the Integrated Rule Inference System (IRIS) on top of a relational database for realizing the storage mechanism and query processing. Finally, we present the overall architecture of the semantic business process repository.

Keywords: Business Process Management (BPM), Business Process Repository, Semantic Business Process Management (SBPM), Semantic Business Process Repository, Ontologies, Reasoning

1 Introduction

The globalization of the economy and the ongoing change of the market situation challenge corporations to adapt their business processes in an agile manner to satisfy the emerging requirements on the market and stay competitive against their competitors. Business Process Management (BPM) is the approach to manage the execution of IT-supported business processes from a business expert's point of view rather than from a technical perspective [SF03]. However, currently businesses have still very incomplete knowledge of and very incomplete and delayed control over their process spaces. Semantic Business Process Management (SBPM) extends the BPM approach by adopting semantic web and semantic web service technologies to bridge the gap between business and IT worlds [HLD+05].

In both BPM and SBPM business processes play a central role. As business processes manifest the business knowledge and logics of a corporation and normally more than one person or organization with different expertise and in different geographic locations are involved in management of business processes, it is necessary to establish a business process repository within the corporation for effective sharing of valuable business knowledge. Furthermore, business users tend to reuse existing business process artifacts during process modeling, so that they are able to adapt the business processes in a more agile manner. However, as the number of business processes increases, it is difficult for them to manage the process models by themselves and to find the required business process information effectively. A business process repository helps business users by providing a systematic way to manage and obtain information on business processes.

In SBPM, business process models are based on process ontologies and make use of other ontologies, such as organizational ontology and semantic web service ontology [HR07]. The business process repository has to be able to cope with these ontological descriptions when storing and retrieving process models, and in particular support efficient querying and reasoning capabilities based on the ontology formalism used. In order to distinguish from traditional business process repository technology, we call this kind of repository a semantic business process repository.

In this paper, we first analyze in section 2 the functional requirements on the semantic business process repository. We describe what kind of functionality the semantic business process repository should offer to its clients, which is primarily a process modeling tool. As a main issue, we identify the integration of a reasoner with the storage mechanism for query processing. In section 3, we then introduce a solution for data storage and query answering based on the Integrated Rule Inference System (IRIS[1]) on top of a relational database. Finally, in section 4, we describe the overall architecture of the semantic business process repository.

2 Requirements Analysis

In general, a repository is a shared database of information about engineered artifacts produced or used by an enterprise [BD94]. In our case these artifacts are semantic business process models. A business process repository has firstly to provide standard functionality of a database management system, such as storage of new business process models, update, retrieval or deletion of existing business process models, transaction support and query capabilities.

The modeling of business processes can be a time-consuming task. It may take days or even months for business users to finish modeling a given business process. Therefore, treating the entire modeling activity related to a business process model as a single transaction is impractical. A semantic business process repository has to provide check-in and check-out operations, that support long running interactions, enable disconnected mode of interaction with the semantic business process repository, and are executed as separate short transactions. In this case the business process modeling tool works in a disconnected mode regarding the semantic business process repository. The semantic business process model in the semantic business process repository is locked when the business process modeling tool obtains it (check-out), so that no other users can modify the SBP model in the meantime. After the modeling

[1] http://sourceforge.net/projects/iris-reasoner/

work has been done, the process model is updated in the semantic business process repository and any locks that have been held for the business process model are released (check-in). Furthermore, in a distributed modeling environment several business users may work on the same process model simultaneously. A fine-grained locking of elements in a business process model enables different business users to lock only the part of the business process model, they are working on, thus avoiding producing inconsistent business process models.

A business process model may undergo a series of modifications undertaken by business users. The series of modifications is called the change history of the business process model. In certain industry sectors corporations must record all the change histories of their business process models for government auditing or for some legal requirements. From the modeling perspective it is meaningful to keep the change histories of the business process models, so that business users can simply go back to an old state in the change history. Therefore, a business process repository must keep change history of each business process model. Each change step in the change history can be represented as a new version of the business process model in the business process repository. A version is a snapshot of a business process model at a certain point in its change history [BD94].

Actually, there are also more other general repository functionalities that a business process repository could provide, such as configuration control, notification service, consistency checking, user management, and security [BD94]. We will, however, not go through these functionalities in detail, because they are not specific to business process modeling.

In SBPM, business process models are enriched by annotating business process artifacts with entities from pre-defined ontologies. There are different kinds of ontologies that are relevant to business process modeling [HR07], such as organizational ontology, Semantic Web Service ontology, business functions ontology, resource ontology, and domain ontologies. In addition, the business process models themselves are modeled in process ontologies. The ontological descriptions of business process models provide a machine-readable representation of business process models and enable machine-processable reasoning on the ontological descriptions. Reasoning can be used for query answering that is based not only on business process artifacts explicitly stored but also on implicit business process artifacts. Besides the functional requirements identified above, a semantic business process repository must integrate a reasoner for query processing in order to exploit the benefit of ontological annotations. The integration of a reasoner for query processing is also what differentiates a semantic business process repository from a conventional business process repository.

The semantic business process repository that we present in this paper stores semantic business process models described using ontologies, which are formalized using WSML-Flight [WSML05]. Therefore, we need a reasoner that can perform reasoning on WSML-Flight. In SBPM, querying the process space of an organization includes not only ontologies for business process modeling but also domain ontologies, ontologies for enterprise data, organizational structure, Semantic Web Services, and business functions, among others. The instances of these ontologies build datasets which are persisted in the underlying data storage and cannot be handled in main memory, because of their size. Typically, reasoners have to load all the data into main memory, and in that case they are not suitable for the semantic business process repository. For query answering in our case the reasoner must be integrated with the storage mechanism and support loading only the required datasets for reasoning.

3 Storage, Reasoning and Query Processing

In the context of the semantic business process repository, storage, reasoning and query processing issues are interrelated. The semantic business process repository stores instances of process ontologies. The use of ontologies enables using reasoning technology to derive implicit knowledge when answering queries. Thus, the query engine which accesses the store has to be integrated with the reasoner. A comparison of different options of storage mechanisms, reasoners, and their integration is out of the scope of this paper. We will describe in the following one possible solution that satisfies the requirements defined in the last section: we use the Integrated Rule Inference System (IRIS) which is integrated with a relational database system.

3.1 Integrated Rule Inference System (IRIS)

IRIS is an inference engine, which together with the WSML2Reasoner framework[2], supports query answering for WSML-Core and WSML-Flight. In essence, it is a Datalog engine extended with stratified negation[3]. The system implements different deductive database algorithms and evaluation techniques. IRIS allows different data types to be used in semantic descriptions according the XML Schema specification and offers a number of built-in predicates. Functionality for constructing complex data types using primitive ones is also provided.

The translation from a WSML ontology description to Datalog is conducted using the WSML2Reasoner component. This framework combines various validation, normalization and transformation functionalities which are essential to the translation of WSML ontology descriptions to set of predicates and rules. Further on, rules are translated to expressions of relational algebra and computed using the set of operations of relational algebra (i.e., union, set difference, selection, Cartesian product, projection etc.). The motivation for this translation lies in the fact that the relational model is the underlying mathematical model of data for Datalog and there are a number of database optimization techniques applicable for the relational model. Finally optimized relational expressions serve as an input for computing the meaning of recursive Datalog programs.

The core of the IRIS architecture, see Figure 1, is defined as a layered approach consisting of:

- Knowledgebase API;
- Invocation API;
- Storage API.

[2] WSML2Reasoner framework: http://tools.deri.org/wsml2reasoner/

[3] IRIS is continuously being developed and the support for non-stratified negation and unsafe rules is envisioned in coming releases.

The knowledgebase API is a top API layer encapsulating central abstractions of the underlying system (e.g., rule, query, atom, tuple, fact, program, knowledge base, context etc.). The purpose of this layer is to define the basic concepts of data model used in IRIS as well as to define the functionality for the knowledge base and program manipulation.

The invocation API characterizes a particular evaluation strategy (e.g., bottom-up, top-down or mixture of these two strategies) and evaluation methods for a given strategy which are used with respect to a particular logic program.

IRIS implements the following evaluation methods[4]:

- Naive evaluation;
- Semi-naive evaluation;
- Query-subquery (QSQ) evaluation.

The storage layer defines the basic API for accessing data and relation indexing. A central abstraction in this layer is a relation which contains a set of tuples and serves as an argument in each operation of relation algebra. The implementation of IRIS relation is based on Collection and SortedSet Java interfaces where red-black binary search trees are utilized for indexing.

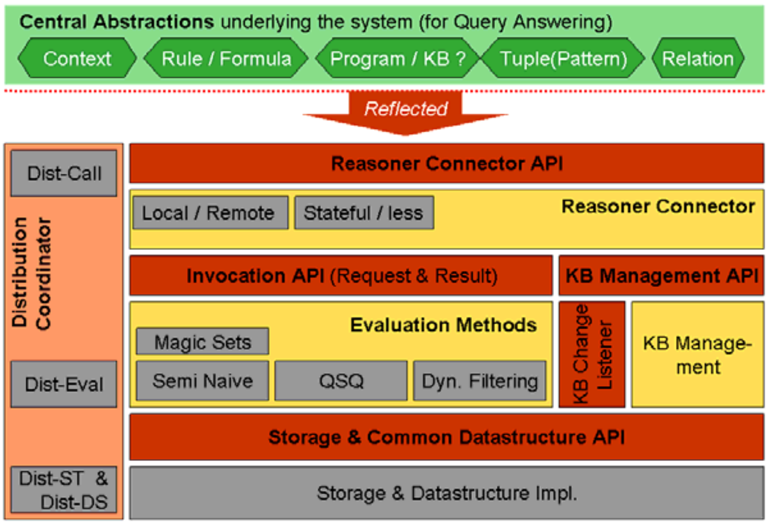

Figure 1: IRIS Architecture

Current inference systems exploit reasoner methods developed rather for small knowledge bases. Such systems either process data in main memory or use a relational database management system to efficiently access and do relational operations on disk persistent relations. Main memory reasoners cannot handle datasets larger than their memory. On the other side, systems based on relational database systems

[4] More evaluation techniques are under development.

feature great performance improvement comparing with main memory systems, but efficient database techniques (e.g., cost-based query planning, caching, buffering) they utilize are suited only for EDB relations and not fully deployable on derived relations.

IRIS is designed to meet requirements for large scale reasoning. Apart from the state-of-the-art deductive methods, the system utilizes database techniques and extends them for implicit knowledge in order to effectively process large datasets. We are building an integrated query optimizer. The estimation of the size and evaluation cost of the intentional predicates will be based on the adaptive sampling method [LN90, RR06], while the extensional data will be estimated using a graph-based synopses of data sets similarly as [SP06]. Further on, for large scale reasoning (i.e., during the derivation of large relations which exceeds main memory), run time memory overflow may occur. Therefore in IRIS we are developing novel techniques for a selective pushing of currently processed tuples to disk. Such techniques aim to temporarily lessen the burden of main memory, and hence to make the entire system capable of handling large relations.

Currently IRIS is a WSML-Flight reasoner. The system is extensively being developed to support reasoning with WSML-Rule (i.e., support for function symbols, unsafe rules and non-stratified negation). Further on, IRIS will tightly integrate a permanent storage system designed for distributed scalable reasoning. One of our major objectives is the implementation of Rule Interchange Format (RIF)[5] in IRIS. Implementing RIF, IRIS will be capable of handling rules from diverse rule systems and will make WSML rule sets interchangeable with rule sets written in other languages that are also supported by RIF.

Finally, IRIS will implement novel techniques for reasoning with integrating frameworks based on classical first-order logic and non-monotonic logic programming as well as techniques for Description Logics reasoning.

3.2 Integration of IRIS with the Semantic Business Process Repository

The semantic business process repository uses a relational database system as the storage mechanism. Relational database systems are widely used both in industry and in research. When using relational database systems, there is no need to re-implement the functionalities such as transaction processing, concurrency control, access control, logging, recovery etc. As relational database systems are widespread for storing data in an organization, using them allows integrating with other enterprise data in a more seamless way.

The needed ontologies, which are formalized in WSML-Flight, are used to generate corresponding relational database schemas. A schema generation tool gets a WSML ontology definition as input and generates the database schema for a particular relational database system (e.g. PostgreSQL[6]) described in SQL Data-Definition Language (DDL).

[5] Rule Interchange Format-W3C Working Group: http://www.w3.org/2005/rules/
[6] http://www.postgresql.org/

When processing queries, the semantic business process repository forwards the query, which is formulated as a WSML logical expression to the IRIS Knowledgebase API. IRIS translates the WSML logical expression to relational algebra statements, from which concrete SQL statements for a particular relational database system are generated.

4 Overall Architecture

In this section, we present the overall architecture of the semantic business process repository. The semantic business process repository has been designed in a layered architecture style consisting of

- Semantic Business Process Repository API;
- Service Layer;
- Persistence Layer.

These three layers are implemented on top of a relational database system. The database schemas are generated from the used WSML ontologies (section 3.2).

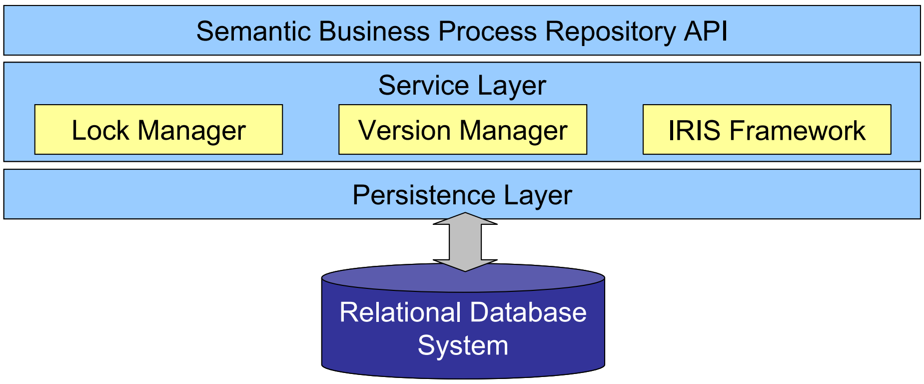

Figure 2: Semantic Business Process Repository Architecture

Semantic Business Process Repository API

The Semantic Business Process Repository API provides programmatic access to the semantic business process repository. It includes an API realizing the CRUD pattern, which represents the four basic functions of persistent storage, namely create, retrieve, update and delete. Besides the CRUD API, the Semantic Business Process Repository API also provides check-in and check-out operations for long-running process modeling. The Query API rounds off the Semantic Business Process Repository API by providing programmatic access to the IRIS Framework for query answering.

Service Layer

The Service Layer implements the Semantic Business Process Repository API and processing logic of the semantic business process repository. The Service Layer contains three modules: Lock Manger, Version Manager and the IRIS Framework. The Lock Manager takes charge of requests on locking and unlocking of the process models in the semantic business process repository. A locking request can only be granted when the process model is not yet locked. The Version Manager takes care of the management of the change histories of process models. To record the change history every new process model or changed process model is stored as a new version in the semantic business process repository. IRIS Framework takes the responsibility for the query processing (section 3.2).

Persistence Layer

The Persistence Layer manages the data access to the underlying relational database system and provides an abstraction for data access operations. It provides persistent solutions for persistent objects by adopting Object Relational Mapping (ORM) middleware such as Hibernate [HIBER] and Data Access Object (DAO[7]) pattern.

5 Summary

In this paper we have presented a semantic business process repository for storage and querying of semantic business process models in SBPM. We have described the functionalities that a semantic business process repository has to provide, namely CRUD API, locking, versioning, and querying using reasoning technology.

In contrast to a conventional business process repository, a semantic business process repository stores instances of process models which are based on ontologies. To exploit the ontological representation, a reasoner has to be used for query processing. Typically, reasoners assume that the whole data is loaded into the main memory, which is not feasible in our case because huge datasets from many different enterprise-related ontologies are needed. As a possible solution we have presented IRIS integrated with a relational database system and described the overall architecture of the repository. We are in the process of integrating IRIS with the relational database system and implementing the semantic business process repository.

Acknowledgements

We would like to thank Marin Dimitrov, Graham Hench, Monika Kaczmarek, Dimka Karastoyanova, Mihail Konstantinov, Tammo van Lessen, Jörg Nitzsche, Jussi Vanhatalo, Karol Wieloch, Paweł Żebrowski and all other colleagues from the SUPER

[7] http://java.sun.com/blueprints/corej2eepatterns/Patterns/DataAccessObject.html

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

References[8]

[BD94] Bernstein, Philip A.; Dayal, Umeshwar: An Overview of Repository Technology. In VLDB 1994.

[HIBER] Hibernate Reference Documentation. Version: 3.2.0.ga

[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 28 - March 2, 2007, Karlsruhe (forthcoming).

[LN90] Lipton, Richard and Naughton, Jeffrey. Query size estimation by adaptive sampling (extended abstract). In PODS ’90: Proceedings of the ninth ACM SIGACTSIGMOD-SIGART symposium on Principles of database systems, pages 40–46, New York, NY, USA, 1990. ACM Press.

[RR06] Ruckhaus, Edna and Ruiz, Eduardo. Query evaluation and optimization in the semantic web. In Proceedings of the ICLP’06 Workshop on Applications of Logic Programming in the Semantic Web and Semantic Web Services (ALPSWS2006), Washington, USA, August 16 2006.

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

[SP06] Joshua Spiegel and Neoklis Polyzotis. Graph-based synopses for relational selectivity estimation. In SIGMOD ’06: Proceedings of the 2006 ACM SIGMOD international conference on Management of data, pages 205–216, New York, NY, USA, 2006. ACM Press.

[SUPER] The European Integrated Project – Semantics Utilised for Process Management within and between Enterprises.
http://www.ip-super.org/

[WSML05] Bruijn, Jos de; Lausen, Holger; Krummenacher, Reto; Polleres, Axel; Predoiu, Livia; Kifer, Michael; Fensel, Dieter: The Web Service Modeling Language WSML. 5 October 2005.
http://www.w3.org/TR/rdf-schema/

[8] All hyperlinks used in this paper were followed on April 10, 2007.

A BPMO Based Semantic Business Process Modelling Environment

Marin Dimitrov[1], Alex Simov[1], Sebastian Stein[2], Mihail Konstantinov[1]

[1] Ontotext Lab. / Sirma Group
135 Tsarigradsko Shose Blvd., Office Express IT Center, Sofia 1784, Bulgaria
{firstname.lastname}@ontotext.com
[2] IDS Scheer AG
Altenkesseler Str. 17, 66115 Saarbrücken, Germany
sebastian.stein@ids-scheer.com

Abstract. The SUPER project presents a novel approach to BPM by using Semantic Web and Semantic Web Services. Existing processes can be augmented with semantic annotations, so that formal reasoning techniques can be applied for discovery, composition, mediation and execution of business processes. This paper introduces a modelling environment that supports the SUPER approach to Semantic BPM.

1 Introduction

The SUPER research project[3] presents a novel approach to Business Process Management by using Semantic Web and Semantic Web Services. Existing business processes can be augmented with semantic annotations and constraints, so that reasoning techniques can be applied for discovery, composition, mediation and execution of business processes. This paper introduces a work in progress on creating a modelling environment that implements the SUPER approach to Semantic Business Process Management.

2 The SUPER Approach to Semantic BPM

Companies have already invested heavily in business process management. For example, they have documented their business processes and created extensive enterprise models. Today, companies are facing problems which can not be solved with current business process management technologies. Therefore, Hepp et al. [1] suggest using semantic technologies like ontologies, mediators, and reasoners.

In order to formally represent business process knowledge, SUPER defines a set of ontologies for business process modelling. The core of the SUPER ontology stack is comprised of five ontologies:

- an *Upper Process Ontology* (UPO), defining top-level concepts such as *task*, *goal* and *condition*

[3] http://www.ip-super.org

- a *Business Process Modelling Ontology* (BPMO), extending the UPO into a full process ontology, providing abstractions over different business process modelling notations such as BPMN [2] and EPC [3]
- *sBPMN* [4], *sEPC* and *sBPEL* [5] – ontologised versions of subsets of the BPMN, EPC and WS-BPEL respectively. sBPEL is additionally enriched with extensions from the Web Services Modelling Ontology (WSMO) [6] for goal-oriented discovery, mediation and execution of services

The SUPER ontology stack provides the means for existing BPMN or EPC models to be ontologically "lifted", i.e. semantically annotated with references to domain ontologies, references to WSMO goals and semantic constraints (in terms of formal pre-conditions, post-conditions, assumptions and effects).

3 A BPMO Based Modelling Environment

3.1 Requirements

The lifecycle of a "minimal" Semantic Business Process Modelling environment has already been outlined in [1]. In summary, the modelling environment should assist the end user in:

- semantic annotation of existing BPMN/EPC process models, i.e. adding references to ontology elements, WSMO goals and semantic constraints. In order to introduce the idea of semantic BPM, it is important to preserve company investments by re-using existing enterprise models. In addition, we don't aim at introducing a new graphical notation but to base our work on what is already used today in industry. Therefore, the semantic modelling tool uses the BPMN notation.
- storing the semantic process models into a Semantic Business Process Library and querying of the library for discovery of existing semantic process models or fragments for reuse, which will decrease the effort and time required for modelling of new processes.
- translation of the semantic business process models represented into executable process models, i.e. a BPMO-to-sBPEL translation. BPMO presents the business user perspective over the business process models and is not sufficient for execution. The modelling environment incorporates a transformation service that will derive executable sBPEL models.

3.2 WSMO Studio

WSMO Studio[4] [7] is an open source, Eclipse based Semantic Web Services modelling environment. With its support for modelling of WSMO elements used in BPMO, such as ontologies, goals and WSML logical expressions (for pre-conditions, post-conditions, assumptions and effects), *WSMO Studio* provides a

[4] `http://www.wsmostudio.org`

good starting point for the BPMO based semantic business process modelling environment in SUPER. Furthermore, its open source licence (LGPL) and Eclipse based architecture, makes it easy for 3^{rd} parties to integrate, customise and extend the provided functionality.

3.3 The BPMO Editor

The BPMO Editor extends *WSMO Studio* with functionality for adding BPMO semantic annotations to existing business process models and for creating new semantic models.

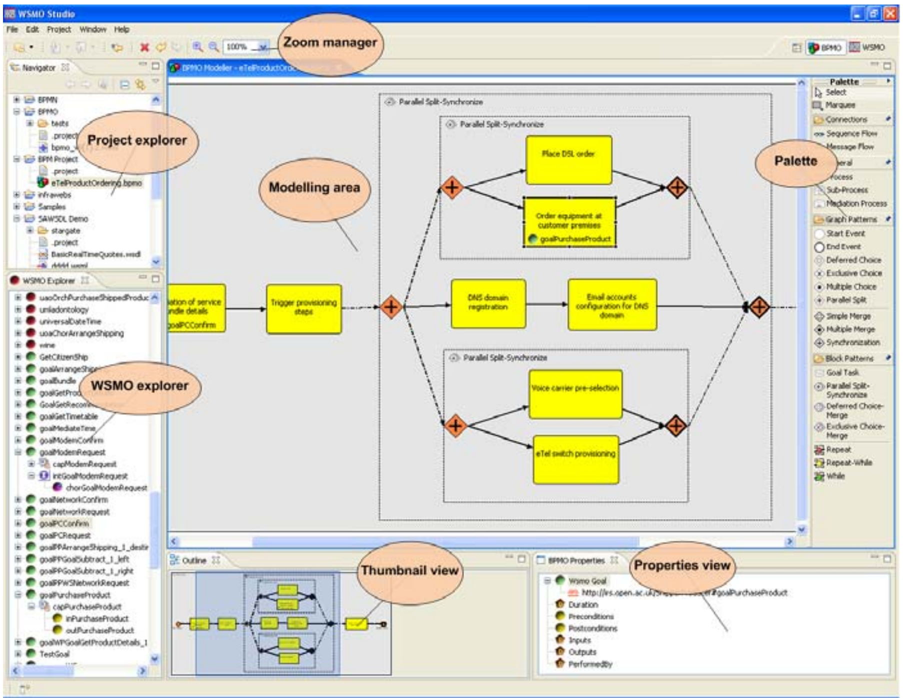

Fig. 1. WSMO Studio with a BPMO editor

The User Interface (Figure 1) is based on the BPMN graphical notation extended with BPMO specific modelling primitives (such as block patterns) and integrated with existing *WSMO Studio* functionality. This way, the end user can simply drag & drop existing semantic elements (e.g. WSMO goals, semantic constraints, concepts and instances from reference ontologies) into the relevant element of the process model (process, activity, data flow elements) to produce the semantically annotated process model in BPMO.

The initial prototype of the BPMO editor is distributed under an open source LGPL licence[5] and available for download at the *WSMO Studio* website.

4 Future Work

The first prototype of the BPMO modelling environment provides the basic functionality for enriching existing process models with semantic annotations, as well as creating new semantic process models from scratch.

Future versions of the prototype will focus on the integration with other components, being developed within the SUPER project, providing means for process validation, discovery, composition and mediation in order to deliver an integrated modelling environment that reduces the human effort required for the translation between the the business needs and IT capability levels.

Acknowledgements

The work described in this document is partly funded by the European Commission under the IST project SUPER (FP6-026850).

References

1. Hepp, M., Roman, D.: An Ontology Framework for Semantic Business Process Management. In: Proceedings of the 8th international conference Wirtschaftsinformatik 2007, Karlsruhe, Germany (2007)
2. Object Management Group: Business process modeling notation specification (BPMN). Technical report, Object Management Group (2006)
3. Scheer, A.W., Thomas, O., Adam, O.: Process modeling using Event-driven Process Chains. In: Process-Aware Information Systems: Bridging People and Software Through Process Technology. Wiley (2005)
4. Abramowicz, W., Filipowska, A., Kaczmarek, M., Kaczmarek, T.: Semantically enhanced Business Process Modelling Notation. In: Proceedings of the Workshop on Semantic Business Process and Product Lifecycle Management (SBPM), Innsbruck, Austria (2007)
5. Nitzsche, J., Wutke, D.: An ontology for executable business processes. In: Proceedings of the Workshop on Semantic Business Process and Product Lifecycle Management (SBPM), Innsbruck, Austria (2007)
6. Roman, D., Lausen, H., Keller, U., de Bruijn, J., Bussler, C., Domingue, J., Fensel, D., Hepp, M., Kifer, M., König-Ries, B., Kopecky, J., Lara, R., Oren, E., Polleres, A., Scicluna, J., Stollberg, M.: Web Service Modeling Ontology, v1.4. WSMO working draft, DERI (2007)
7. Dimitrov, M., Simov, A., Konstantinov, M., Momtchev, V.: WSMO Studio - a Semantic Web Services Modelling Environment for WSMO (System Description). In: Proceedings of the 4th European Semantic Web Conference (ESWC). Number 4519 in LNCS, Innsbruck, Austria (2007) 749–758

[5] `http://www.opensource.org/licenses/lgpl-license.php`

An Ontology-based Modeling Tool for Knowledge-intensive Services

Knut Hinkelmann, Simon Nikles, Lukas von Arx

University of Applied Sciences Northwestern Switzerland, School of Business
Riggenbachstr. 16, 4600 Olten, Switzerland
{knut.hinkelmann, simon.nikles, lukas.vonarx }@fhwn.ch

Abstract. The ATHENE modelling tool enables business people to create knowledge-intense process models without having to know the complexity of modelling ontologies. ATHENE is based on a three-level hierarchy of meta-meta, meta and object level. It allows business people to model business processes graphically meanwhile these processes are transformed internally into ontologies.

1 Objectives

Modelling business processes as ontologies is cumbersome and requires in-depth expertise of semantic technologies. However, business process modelling is a task mainly for business people who lack this expertise. So, the fundamental problem is that traversing from one sphere to the other requires manual labour in any of the two directions, i.e. both for querying and manipulation the process space [4]. To overcome this difficulties we developed a system called ATHENE for graphical modelling of business processes that are automatically transformed into ontologies. It corresponds to the theories of the Semantic Business Process Management as described by Hepp et. al. [4] which combines semantic web services (SWS) and Business Process Management (BPM).

From the user interface the modelling tool is equivalent to any other business process modelling tool. The business expert can model business processes in a familiar way. Internally, however, these models are represented as an ontology which results in a semantic representation of the process.

The idea of modelling in ATHENE is illustrated in figure 1. On the meta-level there is an ontology defining the concepts and properties for business process and service modelling. Examples of these ontologies could be OWL-S [6] or WSMO [8]. OWL-S for example contains concepts for

- atomic and composite services,
- control constructs like sequence, split+join, if-then-else, iteration, while, switch, ...[6]

In our ATHENE modelling tool we define a graphical representation for each of these concepts. In addition, for each modelling object an interface is defined so that a user can specify properties. In our tool this interface is called “notebook”.

2 Meta-meta Modelling approach

Business process modelling, however, not only consists of modelling the processes themselves. In addition, there are other dimensions like organisational structure or data models for which there should be own model types, each consisting of ontology concepts with associated graphical representations.

To support the definition of new model types, our ATHENE system is based on a meta-meta-modelling approach, resulting in a three-level model hierarchy. The meta-meta level (also called meta^2-level) specifies the basic constructs for defining a model type, i.e. it predefines and allocates classes for the meta-level beneath.

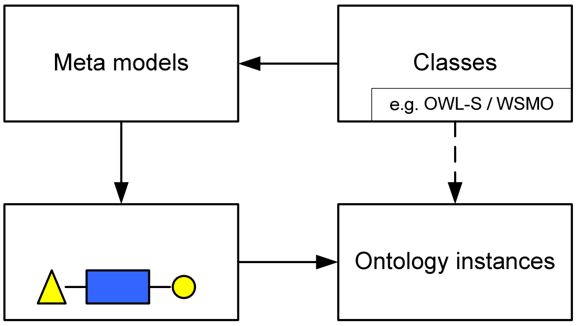

Fig. 1. The idea of modelling in ATHENE

The meta^2-modelling approach of ATHENE not only allows to easily define new model types but also to adapt the modelling environment for any kind of process modelling notations (e.g. BPMN), data models (e.g. ERM or UML), organisational structures or ontologies themselves. Thus, ATHENE can be regarded as a user-friendly graphical environment to model organizational structures which internally are represented as enterprise ontlologies, similar to the proposal of the TOVE Enterprise Modelling Project [8]. This offers a big flexibility as well as the possibility to adapt the modelling environment to a certain modelling notation. As ATHENE stores all information in a semantic manner and allows the modelling of any notation it is possible to generate Enterprise Models and combining different notations (e.g. process models, rulesets and ontologies). ATHENE could therefore be seen as first step towards SBPM as proposed by Hepp [4]. These combination options might also be a helpful for approaches like DEMO shown by Diez [3] where actors could be seen as organizational units connected with transactions.

3 The meta^2-level

The simple meta^2-model of Sinz [7] consists only of meta-object-types and meta-relations. There are three basic relations inside a meta^2-model: "is a", "has" and "connects". On the other hand, the Adonis modelling tool has a more complex, object-oriented meta^2-model [5]. For ATHENE we defined a meta^2-model that is generally based on the definitions of the meta^2-model of Sinz [7] but with useful extensions. Although, it does not have the complexity of the meta^2-model of Adonis as it has to be held more flexible.

4 Implementing meta^2-level

There are two different ways to implement meta^2-models. On one hand a meta^2-*model represented in a programming language* and on the other hand a *meta^2-model explicit expressed*. The former leads to an fixed meta-model where the adaptation is only possible via predefined model-, object- or attribute-types whereas the latter offers the possibility to create user-specific meta-models based on the definitions on the meta^2-level [2].

For ATHENE a meta^2-model which is expressed explicitly is much more suitable. Although, this kind of meta^2-model offers a bigger flexibility, it has to be considered that a meta^2-model that has been defined and used on meta-level cannot be changed at a later stage. This is due to the fact that meta-models are always based on a specific meta^2-model. A later change at the meta^2-model might lead to inconsistency and must therefore be prohibited.

But in spite of the theoretic complexity, one of the main goals is to make ATHENE user friendly what includes a way to easily create any kind of new meta-model on meta-level. The meta^2-level defines and allocates classes for the meta-levels beneath [1]. To realise this, the meta^2-model has to be comprehensive in knowledge covering.

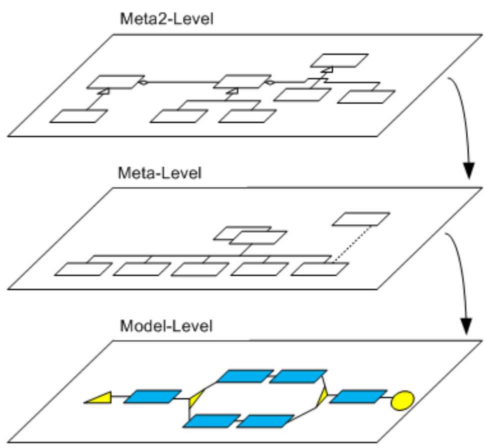

Fig. 2. The three-level-hierarchy

However, to put it in a nutshell, the tree-level-hierarchy works like this:

On top-level there is the meta^2-model. It defines the meta-object-types meta-relations and attributes represented as OWL classes. This meta^2 ontology defines guidelines for the meta layer. According to these definitions it is possible to define user-specific meta-models on the meta-level by specifying subclasses of the classes defined on the meta^2 ontology. In the end, concrete process models, data models or structures are modelled as instances of the classes of the meta language. *Figure 2* illustrates this architecture schematically.

5 The ATHENE system

ATHENE is implemented as web based application, what allows users to work in several places and share (meta-)models without exchanging files while no software has to be installed. Because of different strengths and advantages such as maturity, reliability, power and its similar behaviour in different browsers, Java Applet technology was applied.

To facilitate extensibility and optimize load time, ATHENE is built as a plug-in oriented framework where components (e.g. to define a model type) are loaded on demand. New components and subcomponents can be developed independently and made available in the base application through parameterisation.

References

1. Antoniou G.; van Harmelen, F. (2004): Semantic Web Primer, p.211, The MIT Press, Cambridge, Massachusetts
2. Becker, J.; Kugler, M.; Rosmann, M. (2001): Prozessmanagement, Münster/Brisbane
3. Dietz, J.L.G., The deep structure of business processes, Communications of the ACM, Volume 49 , Issue 5 (May 2006), pp. 58 – 64
4. Hepp, M. et. Al.: Semantic Business Process Management: A vision Towards Using Semantic Web Services for Business Process Management
5. Junginger, S. et al. (2000): Ein Geschäftsprozessmanagement-Werkzeug der nächsten Generation – ADONIS®. In: Wirtschaftsinformatik, 42. Jg. 2000, Heft 5, S. 392 – 401
6. Martin, D. et al. (2004): OWL-S: Semantic Markup for Web Services, URL: http://www.w3.org/Submission/OWL-S
7. Sinz, E. J (1996).: Ansätze zur fachlichen Modellierung betrieblicher Informationssysteme – Entwicklung, aktueller Stand und Trends. In: Heilmann, H., Heinrich, L.J., Roithmayr R.: Information Engineering, Munich, Vienna.
8. http://www.eil.utoronto.ca/enterprise-modelling/tove/index.html
9. www.wsmo.org