SWB Process
A Business Process Management System driven by Semantic Technologies

Hasdai Pacheco, Karen Najera, Hugo Estrada and Javier Solis
Fund of Information and Documentation for the Industry INFOTEC, Mexico DF, Mexico

Keywords: Business Process Management System, Semantic Technologies, Ontology-Driven Development.

Abstract: Constant changes in the market force enterprises to continuously define and redefine their business processes, and the technology that supports them, in order to fulfill the organizational objectives. Business Process Management Systems (BPMS) are intensively used in organizations as a useful tool to face those changes. However, as stated in literature and practice, a BPMS still has to cope with the agility to adapt to changes and the low degree of automation of the BPM life-cycle. These issues have been faced through the integration of BPM with Semantic Technologies. Some proposals are focused in conceptual approaches while others involve tools to cover part of the BPM life-cycle. Nevertheless, there are no works that have implemented a BPMS that exploits Semantic Technologies for covering the whole BPM life-cycle. In this paper we present SWB Process, an industrial and Open Source BPMS completely driven by Semantic Technologies that uses ontologies to agilely support constant changes in the processes of organizations, increasing the degree of automation of the BPM life-cycle. Moreover, we take advantage of ontologies to quickly adapt it to new BPM needs. SWB Process has been validated through real projects in several government agencies in Mexico.

1 INTRODUCTION

Constant changes in the market force enterprises to continuously define and redefine their business processes, and the technology that supports them, in order to fulfill the organizational objectives. In this context, the paradigm of Business Process Management (BPM) has been widely accepted in industry and research to optimize enterprise resources and core activities, since it encompasses methods, techniques and Information Technologies (IT) to manage business processes involving humans, applications, documents and other sources of information (van der Aalst et al., 2003). BPM is directed by a life-cycle that comprises four phases: *modeling*, *implementation*, *execution*, and *analysis* (Wetzstein et al., 2007). These phases can be covered by systems known as Business Process Management Systems (BPMS). Thus, a BPMS allows to generate business processes models and provides mechanisms to translate those models into an executable system that helps process performers to accomplish their business tasks.

Several BPMS have been proposed in academy and industry (Butti et al., 2013; Domingue et al., 2013; Jain et al., 2013; Calkins et al., 2013). However, as stated in literature and practice (Hepp et al., 2005; Wetzstein et al., 2007; Filipowska et al., 2011), there are still some issues that a BPMS has to address, such as the agility to adapt to changes and the low degree of automation of the BPM life-cycle. In order to overcome these issues, researchers are tackling the integration of BPM with Semantic Technologies since this integration offers inherently more flexibility for supporting and increasing the degree of automation of the BPM life-cycle in changing scenarios (Davis, 2005; Filipowska et al., 2011). Some proposals are focused in conceptual approaches, that is, BPM ontologies and formalizations (Panos and Gómez, 2012; Oro and Ruffolo, 2012; Mueller, 2012), while others involve tools using ontologies to cover part of the BPM life-cycle, or architectures and functional requirements for a Semantic BPMS (Wetzstein et al., 2007; Karastoyanova et al., 2008; Domingue et al., 2013). Nevertheless, there are no works that have implemented a BPMS that exploit Semantic Technologies for covering the whole BPM life-cycle.

In this paper, we present an industrial and Open Source BPMS called SWB Process[1] completely driven by Semantic Technologies (ontologies, triplestores, query languages and reasoners) that supports the whole BPM life-cycle. SWB Process has been de-

[1] http://www.semanticwebbuilder.org.mx/SWBProcess

Pacheco H., Najera K., Estrada H. and Solis J..
SWB Process - A Business Process Management System driven by Semantic Technologies.
DOI: 10.5220/0004714705250532
In *Proceedings of the 2nd International Conference on Model-Driven Engineering and Software Development* (MODELSWARD-2014), pages 525-532
ISBN: 978-989-758-007-9

veloped following the Ontology-Driven Information Systems (ODIS) approach (Guarino, 1998; Uschold, 2008). Accordingly, ontologies drove the development of SWB Process, and they also play an important role during the supported BPM life-cycle. By using ontologies, SWB Process provides a solution to agilely support constant changes in organizations increasing the degree of automation of modeling, implementation, execution, and analysis phases of BPM life-cycle. Moreover, we take advantage of Ontology-Driven Development, such as maintenance and update tasks improvement, making SWB Process flexible to be quickly adapted to new BPM needs.
SWB Process has been successfully validated through real projects, supporting the business processes of several government agencies in Mexico.
The paper is structured as follows: Section 2 describes the development of SWB Process. Section 3 presents how SWB Process supports BPM life-cycle along with an example. Section 4 describes the benefits of SWB Process. Section 5 presents related works. Finally, section 6 presents our conclusions and ongoing work.

2 SWB PROCESS DEVELOPMENT

In this section, the development of SWB Process is presented. SWB Process is a Semantic BPMS that uses the Business Process Model and Notation (BPMN) 2.0 (OMG, 2011) as business processes modeling language and a Web architecture for business process execution and deployment. Ontology-Driven Information Systems ideas (Guarino, 1998; Uschold, 2008) were followed to define SWB Process. Therefore, its development was addressed by using an Ontology-Driven Development Framework called SemanticWebBuilder (SWB) (Solis et al., 2013), which provides the mechanisms to generate the base source code of Web applications starting from an ontology that defines system requirements.
For the development of SWB Process we used the following components provided by SWB: a) A domain ontology, defined with the Web Ontology Language OWL (OMG, 2004a), that describes system requirements for Web applications (SWBOntology); b) a code generator to transform ontology definitions (extending from the SWBOntology) into Java source code (SWBCodeGen); and c) a platform with libraries and utilities to accelerate software development and to encourage software reuse (SWBPlatform).
We describe, in the following subsections, the three major steps followed to generate SWB Process (Fig. 1): *Modeling the SWB Process Ontology*, *Automatic code generation* and *Specific development*.

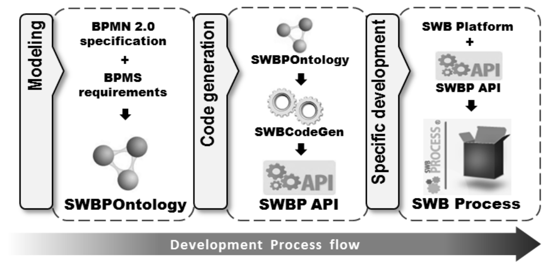

Figure 1: SWB development methodology.

2.1 Modeling the SWB Process Ontology

The SWB Process ontology[2] (SWBPOntology), is an OWL ontology that captures the functional requirements of SWB Process, as well as the relevant concepts and behavioral aspects from the Business Process Model and Notation (BPMN) 2.0 specification (OMG, 2011). For the SWBPOntology modeling, two previous activities were achieved: First, the identification of relevant classes from the BPMN specification and second, the identification of concepts and behaviors coming from system requirements for the implementation of the final BPMS.
For the first activity, elements for process modeling and execution coming from the BPMN specification were considered (expressed as UML class diagram fragments). This included the elements involved only in BPMN process orchestration and BPMN collaborations i.e. omitting those elements for BPMN choreographies and BPMN conversations. As a result, we had a set of candidate BPMN classes from the Common, Foundation, Collaboration, Activities, Artifacts, Events, Data, Gateways and Process packages. Examples of these classes are the Activity, Process, Event, Subprocess, CallActivity, Gateway, BaseElement, Documentation and Task classes, among others. It is important to point out that as our intent was to provide process model persistence and data interchange through RDF and OWL formats, the section on BPMN Diagram Interchange and Exchange formats of the specification were not considered in the candidate classes identification.
For the second activity, the gathering of additional requirements for the BPMS implementation was needed because the BPMN specification is mainly focused in defining the notation and semantics for process modeling and interchange using XML, thus it lacks of

[2] http://www.semanticbuilder.com/SemWB4/SWB4/swbp/WEB-INF/owl/ext/swp.owl

implementation details for a final BPMS. This activity led us to the identification of ontology concepts and behaviors needed to support the BPM life-cycle through the BPMS components. Examples of the identified concepts are, service concepts (such as Web Service (WS), SPARQL querying and DataBase querying), documents, templates, user interaction components and process execution concepts (such as process instance and flow node instance).
With the candidate classes of the BPMN specification and the concepts from the BPMS requirements the SWBPOntology modeling was carried out extending the SWBOntology provided by the SWB Framework (Solis et al., 2013). Thus, OWL classes and properties corresponding to the identified concepts and candidate classes were created. We reused the SWBOntology since it defines classes and properties to support the automatic transformation from the ontology to Java source code and concepts for Web components and Web site features development needed since SWB Process performs process execution in a Web architecture.
The main challenge in the modeling step was the mapping between the BPMN class structure (which includes multiple inheritance of classes) and separated OWL classes (concepts and behaviors) in the SWBPOntology as required by the SWB Framework (where multiple inheritance is allowed only through behaviors). At this point, the SWBOntology played a fundamental role, since it includes definitions useful to fill the gaps in the implementation of the BPMN specification, for instance, participants are defined in BPMN as roles or resources but only at conceptual level, on the other hand, SWBOntology has a definition of roles and users that can be reused by SWB Process along with their source code for user management, user registration and user validation.
At the end of the modeling step, a total of 164 OWL classes for the SWB Process implementation were defined in the SWBPOntology, of which 151 are concept definitions and 13 are behavioral concepts.

2.2 Automatic Code Generation

This step consisted in the generation of SWB Process base source code (in Java programing language) by executing the SWBCodeGen. Therefore, it was configured to use the SWBPOntology as input to get as output the base source code of SWB Process, which corresponds to a domain-specific and high-level Java API (SWBP API). The generated SWBP API encapsulates the source code (classes and methods) necessary to achieve the persistence of the objects involved in the different components of SWB Process, including: 1) a set of Java classes and interfaces corresponding to the OWL concepts and behavioral aspects defined in the SWBPOntology; and 2) a set of class methods to access the corresponding OWL properties defined in the SWBPOntology.
The SWBP API (comprised by 302 java classes and 13 java interfaces) is supported by the SWBPlatform which provides a set of libraries that allows developers to use connectors to several triplestores for RDF storage (such as Bigdata and Apache Jena). In this way, the SWBP API helps developers to reduce the complexity of managing RDF persistence in a standardized way in which data persistence mechanisms are separated from the business logic of the end application. Thus, accelerating the application development.

2.3 Specific Development

This step consisted in the development of the components and the operational business logic by using the SWBP API to cover the SWB Process functionalities. The operational business logic of each BPMN element was developed taking into account the BPMN execution semantics from the BPMN 2.0 specification and previous experience in the development of workflow systems. This execution semantics served as the basis for the definition and implementation of a state-based process execution engine, as well as user interaction components for the modeling, configuration and management of business processes. These components, whose architecture is shown in figure 2, allow end users to manage the BPM life-cycle in a generic way.
The SWB Process components are described below.
Process Modeler (SWBP Modeler). The SWBP Modeler is a Web based component that supports to graphically design business processes by using BPMN 2.0 notation taking into account the BPMN execution semantics (e.g. which BPMN elements can be connected in a process flow). It includes a mechanism to relate ontologies with a business process to define the structure of business artifacts, such as data, documents or Web Services. The SWBP Modeler maps in a transparent manner to the user, each graphical element of a particular business process to its corresponding concept in the SWBPOntology, generating as output a SWBPOntology instance which defines a *Semantic Process Model*.
Configuration and Deployment Module (SWBP Configurator). This component provides a Web based user interface that takes as input the *Semantic Process Model* for its configuration in order to make it executable and deployable on a Web site. The config-

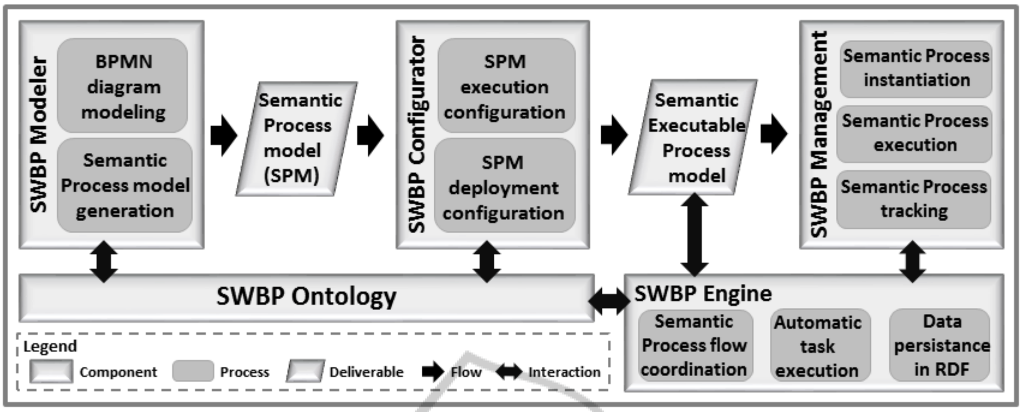

Figure 2: SWB Process Architecture.

uration comprises two parts: a) the execution configuration, which consists of capturing the values of each property of the graphical elements of the process. The properties are those defined in the SWBPOntology belonging to the *Semantic Process Model*. Moreover, it includes the relation of BPMN data objects defined in the *Semantic Process Model*, with classes of the SWBPOntology or other ontologies that describe business artifacts used along the process flow. b) the deployment configuration, which consists of the definition of Web form templates of process activities for user interaction. The output of this component corresponds to the *Semantic Executable Process Model*.

Business Process engine (SWBP Engine). The business process engine empowers process execution and monitoring. It implements a state machine that coordinates business process flow taking into account properties, behavior and constraints defined in the SWBPOntology for each BPMN element, as well as all the specific configuration defined in the *Semantic Executable Process Model*. It also manages the execution of Service and Script tasks. Moreover, the SWBP Engine manages the persistence of all data of the process and its execution in RDF format.

Management Module (SWBP Management). This module allows the instantiation of the *Semantic Executable Process Model* to generate *Semantic Process instances*. It provides a business task inbox to accomplish the execution of *Semantic Process instances*. The business task inbox allows users to perform their tasks and to reallocate human resources for the tasks. Moreover, the SWBP Management provides a module for process tracking that retrieves process instance data (stored in RDF format by the SWBP Engine) and generates tables and graphs to show *Semantic Process instances* execution performance. Furthermore, it provides an SPARQL Endpoint to query process information, not only from a process instance, but also from the process structure itself, such as, process flow.

3 SWB PROCESS AND BPM LIFE-CYCLE

In this section is described how Semantic Technologies, as basis of SWB Process, support the BPM life-cycle. To do this, first, we present the definition proposed in (Wetzstein et al., 2007) for each phase of the BPM life-cycle. Then, we explain how SWB Process gives support to the phases.

- *Modeling:* in this phase, business analyst creates a business process model with help of a modeling tool by specifying the order of tasks in the business process. This phase is covered by SWB Process through the SWBP Modeler, which supports the design of BPMN diagrams and transforms those diagrams into a *Semantic Process model*.
- *Implementation:* in this phase, the business process model, created in the modeling phase, is transformed and enriched by IT engineers into an executable process model. This phase is covered by the SWBP Configurator which leads the configuration of the *Semantic Process model* (generated in the modeling phase) for its execution and deployment on a Web site. After the configuration, the *Semantic Process model* becomes a *Semantic Executable Process model* which is ready to be executed.
- *Execution:* in this phase, a process engine executes a process instance (an specific execution of the executable process model), by navigating through the control flow of the executable process model. This phase is covered by SWB Process through the SWBP Engine and the SWBP Management. The SWBP Engine empowers the *Se-*

mantic Process instance execution coordinating the process flow, taking into account the configuration defined in the *Semantic Executable Process model*. The SWBP Management manages the execution of *Semantic Executable Process models* on a Web site, for instance, generating new *Semantic Process instances* and providing a business task inbox to list tasks to be performed by a user.

- *Analysis:* in this phase, process analysis comprises monitoring of running process instances and process mining. This phase is covered by SWB Process through the SWBP Management. It includes a tracking component that displays: process execution performance and information of the running *Semantic Process instances*. Moreover, an SPARQL Endpoint is provided for querying process information, such as, process flow or *process instance* execution and data.

3.1 SWB Process in Practice

In this section, the workflow with SWB Process is presented, which consists of seven steps that cover the BPM life-cycle. The first five steps must be performed by the business analysts or engineers to define a *Semantic Executable Process Model* along the modeling and implementation phases, the sixth and seventh steps are useful during execution and analysis phases, where process orchestration is performed. Following, an example of a process generated with SWB Process is described that comprises the mentioned steps. The example corresponds to the abstract of one of the processes implemented in our research center (INFOTEC), whose objective is to manage the employee vacation requests. The process flow is as follows:

An 'employee' sends a vacation request, the 'dept. supervisor' can approve the vacation request, reject it, or ask the 'employee' to reschedule dates. In case the 'dept. supervisor' approves the vacation request, 'human resources' department has to validate it. If 'human resources' department validates the request, the 'employee' is notified via e-mail about the approval, otherwise, he is notified about the rejection; in case the 'dept. supervisor' rejects the request, the 'employee' is notified via e-mail about the rejection; and finally, in case the 'dept. supervisor' ask for rescheduling dates, the 'employee' may modify dates and send the vacation request again.

Process participants are: *employee*, *dept. supervisor* and *human resources department*. Whereas data involved in the process are: vacation start date, vacation end date, request comment, reject comment, vacation

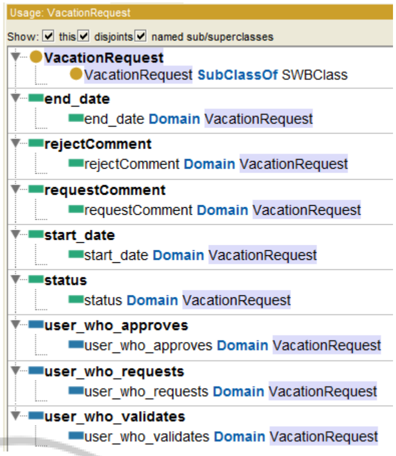

Figure 3: Vacation request data modeled in Protege.

request status and validation.

As a first step, business artifacts are defined in terms of ontologies. In the example we refer to process participants and data. For this purpose, an ontology editor such as Protege[2] or TopBraid[3] can be used. We have taken the SWBOntology (Solis et al., 2013) as basis to reuse its User definition for process participants, therefore, a new ontology that extends the SWBOntology is generated. On this ontology, the VacationRequest class was created to define data as data type properties of the class. The object properties: user_who_request, user_who_approves and user_who_validates, were created to relate the User definition in the SWBOntology with the VacationRequest class. Fig. 3 shows the implementation of the VacationRequest class in Protege. The second step corresponds to graphically model the BPMN diagram, which includes the business process flow and business rules by using the *SWBP modeler*. Fig 4 shows the Vacation request process (the database symbol located at the bottom represents the association of the VacationRequest class defined in the ontology). In the third and fourth steps, participants and data are related with process activities and execution properties are configured for each process element, this includes the association of ontology classes to the process data objects. The fifth step is related to the definition of Web form templates of process activities for user interaction. In Fig. 5, the *employee request task* configuration is presented. Data defined in the ontology and related to process data objects is listed and the user can select which property will appear in the Web form template for this task, in this case: start date, end date and request comment from the VacationRequest class defi-

[2]Protege. http://protege.stanford.edu/

[3]TopBraid. http://www.topquadrant.com/products/

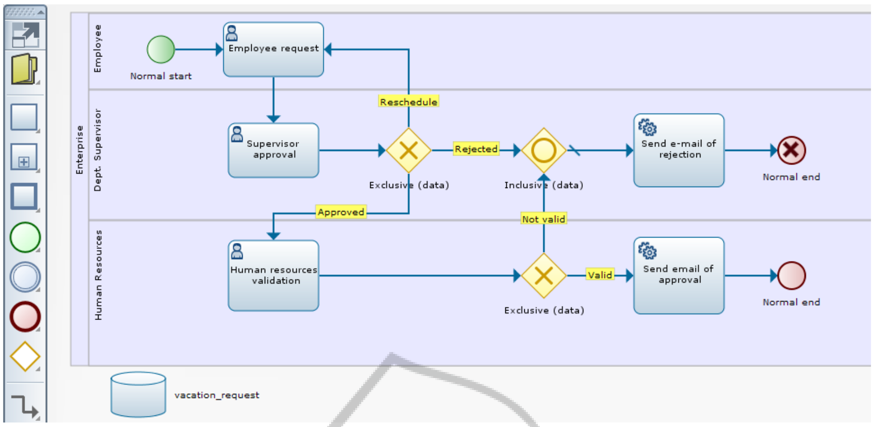

Figure 4: Vacation request process modeled with SWB Process modeler.

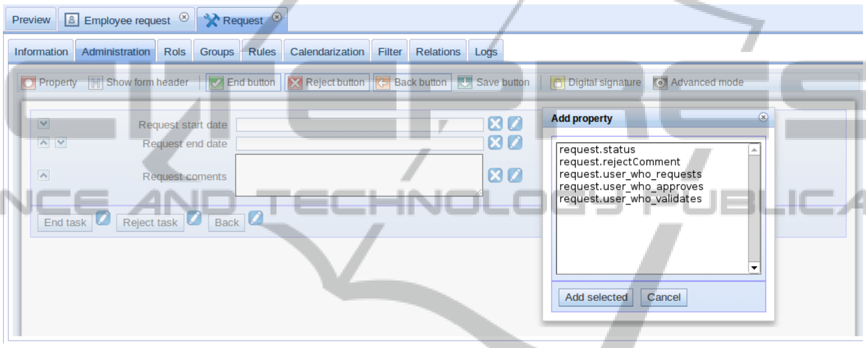

Figure 5: Employee request task configuration.

nition. Moreover, the user can define the type of form element to be used for each concept property, such as Text area or calendar date selector. After the fifth step, the process is ready to be deployed in a wrapper Web page. In the sixth step, the *employee* participant can generate a process instance to perform a vacation request through the User task inbox of the SWBP Management. Fig. 6 displays the *employee request task* deployment.

Finally, in the seventh step, monitoring, tracking and querying process data can be performed through the SWBP Management.

4 SWB PROCESS BENEFITS

Using an Ontology-Driven approach in the development of SWB Process allowed us to provide advantages over other industrial BPMS in several aspects. In terms of changes in requirements, ontologies provide a flexible way to adapt to new BPM needs, helping to increase the automation level of the BPM life-cycle. For example, if the BPMN notation changes and more primitives are added, it is possible to quickly adapt SWB Process including the new primitives in the SWBPOntology and in consequence, in the SWB Process source code. The same is true for changes in the execution semantics of the BPMN specification or for changes in the way the BPM life-cycle works on business processes. Moreover, using ontologies to define the system architecture and data model enables the interchange of process definition and execution information between other components, agents or IT systems in a standard machine-readable format (OWL and RDF (OMG, 2004a; OMG, 2004b)) without the need for additional data treatment. In terms of dynamic workflow management, ontologies in SWB Process provide a mechanism to change process definitions and execution properties on the fly, maintaining the consistency of the running process cases without the need of process model (re)compilation. This is possible because all process models are a set of individuals of the SWBPOntology and the SWBP Engine acts as a model interpreter for the process models, instead of an application compiler for each model.

On the other hand, by using ontology concepts as data structures for business objects configuration in a business process model, SWB Process provides a way to make explicit the knowledge involved in business

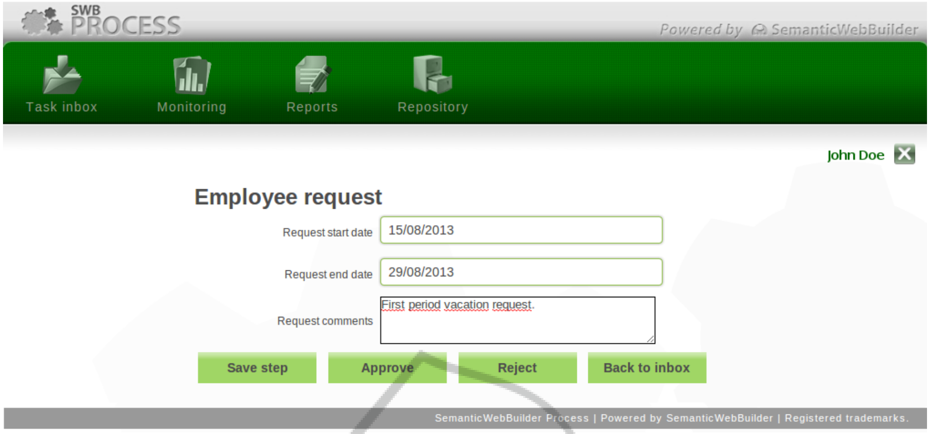

Figure 6: Employee request task deployment.

process execution, enabling automatic process space querying and reasoning tasks over process execution data for business analysis, intelligence and improvement. Besides these benefits, as business processes are described with Semantic Web standards, interoperability between business processes is enabled. Furthermore, business process knowledge can be published through a SPARQL endpoint on the Semantic Web, and other Semantic Web paradigms such as Linked Data can be applied to exploit business knowledge and to enrich business process definitions.

5 RELATED WORKS

Research works are tackling the integration of BPM with Semantic Technologies from different perspectives. Some proposals are focused in theoretical and conceptual approaches, for instance: in Oro work (Oro and Ruffolo, 2012), a framework to create business process ontologies is presented. Ontologies can be queried and exploited to monitor process models, extract information from documents, execute processes and monitor the execution, and finally, analyze process instances. The work of Mueller (Mueller, 2012) provides three ontologies with important concepts of existing BPMS, such as classes and properties of the BPMN 2.0 specification and the Service Component Architecture (SCA) assembly model. Some applications of this ontologies in the BPM life-cycle are described.
On the other hand, some approaches are focused on the definition of system requirements for a Semantic BPMS, or involve tools to partially cover the BPM life-cycle, for instance: the work of Wetzstein (Wetzstein et al., 2007) describes functional requirements for a Semantic BPMS, according to the BPM life-cycle: modeling (semantic annotation, and process fragments), implementation (process composition), execution (dynamic SWS discovery and invocation) and analysis (process mining and monitoring). The work of Karastoyanova (Karastoyanova et al., 2008) presents a reference architecture for a Semantic BPMS. The architecture comprises functionalities for each phase of the BPM life-cycle. Moreover, the authors propound a Semantic Execution Environment and show how existing BPMS components can be extended with semantic features. The SUPER Project (Domingue et al., 2013) applies semantic technology to acquire, organize, share and use the stakeholders knowledge and knowledge embedded in business processes within existing IT systems, in order to make companies more adaptive.

However, there are no works that have implemented a BPMS that exploit Semantic Technologies for covering the whole BPM life-cycle, which is the case of our approach. Our BPMS encompasses Semantic Technologies at conceptual and implementation levels. At conceptual level, our BPMS has an ontology (SWBPOntology) that defines elements of the BPMN 2.0 specification along with BPMS system requirements (similar to Mueller (Mueller, 2012)), moreover, additional ontologies can be created and associated to specific business processes to define business objects making explicit the knowledge involved in a business process. In contrast to Oro work (Oro and Ruffolo, 2012), where a framework is provided to create process ontologies, in our BPMS, business processes are automatically represented in terms of ontologies, when they are modeled with BPMN 2.0, since business processes are SWBPOntology instances. Thus, having the same ontologies advantages without needing to generate them separated nor giving process data treatment. At implementation level, we present an implemented BPMS that covers the whole BPM life-cycle, in contrast to

SUPER (Domingue et al., 2013) which is focused on Semantic Web Services, and in contrast as well to Wetzstein (Wetzstein et al., 2007) and Karastoyanova (Karastoyanova et al., 2008) works, where BPMS functional requirements are proposed without system implementation.

6 CONCLUSIONS

In this paper, we have presented an industrial and Open Source Semantic BPMS called SWB Process. It has been developed following the Ontology-Driven Information Systems approach. Accordingly, ontologies were directly involved in the development of SWB Process through Ontology-Driven Development, and ontologies also play an important role during the supported BPM life-cycle. By using ontologies as the basis of SWB Process, we provide a solution with flexible and agile mechanisms to adapt to new BPM needs and continuous changes in organizations, increasing the degree of automation and better supporting the BPM life-cycle. Moreover, the information implicitly represented in a business model has explicit meaning, therefore, machines as well as people are enabled to understand, share and reason over business processes models and information. In addition, other Semantic Web paradigms can be applied to exploit business information such as Linked Data.

SWB Process has been successfully implemented and validated through real projects, supporting the business processes of several government agencies in Mexico, for instance, the Federal Electricity Commission (CFE)[4] and the National Institute of Women (INMUJERES)[5]. Moreover, it has been used to implement processes of our research center (INFOTEC).

In tandem with SWB Process, we provide the following support services: consultancy, mentoring, technical support, training and customization.

In our ongoing work, we are addressing the semantic annotation of business processes with external ontologies, to clarify processes through generic concepts, providing additional support to business analysis during modeling and enabling reuse of information.

REFERENCES

Butti, S. et al. (2013). WebRatio tool. Online. http://www.webratio.com/. Last access: 05/07/2013.

Calkins, M. W. et al. (2013). Apian tool. Online. http://www.appian.com/. Last access: 05/07/2013.

[4]http://www.cfe.gob.mx/

[5]http://www.inmujeres.gob.mx/

Davis, M. (2005). The Business Value of Semantic Technology, from vision to mainstream markets 2000 - 2010. In *Semantic Web Applications for National Security*.

Domingue, J. et al. (2013). SUPER Project. Online. http://www.ip-super.org/. Last access: 05/07/2013.

Filipowska, A., Kaczmarek, M., Koschmider, A., Stein, S., Wecel, K., and Abramowicz, W. (2011). Social software and semantics for business process management - alternative or synergy? *Journal of Systems Integration*, 2(3):54–69.

Guarino, N. (1998). Formal ontology and information systems. In *Proceedings of Formal Ontology in Information System*, pages 3–15. IOS Press.

Hepp, M., Leymann, F., Domingue, J., Wahler, A., and Fensel, D. (2005). Semantic business process management: a vision towards using semantic web services for business process management. In *e-Business Engineering, ICEBE 2005*, pages 535–540.

Jain, R. et al. (2013). Intalio tool. Online. http://www.intalio.com/. Last access: 05/07/2013.

Karastoyanova, D., Lessen, T. V., Leymann, F., Ma, Z., Nitzsche, J., Wetzstein, B., Bhiri, S., Hauswirth, M., and Zaremba, M. (2008). A reference architecture for semantic business process management systems. In *Multikonferenz Wirtschaftsinformatik*.

Mueller, R. (2012). Enterprise applications of semantic technologies for business process management. *Journal of Zhejiang University SCIENCE C*, 13(4):308–310.

OMG (2004a). OWL Web Ontology Language. Online. http://www.w3.org/TR/2004/REC-owl-features-20040210/.

OMG (2004b). Resource Description Framework (RDF). Online. http://www.w3.org/RDF/.

OMG (2011). Business Process Model and Notation 2.0. http://www.omg.org/spec/BPMN/2.0/.

Oro, E. and Ruffolo, M. (2012). A knowledge representation formalism for semantic business process management. In *Advances in Knowledge Representation*.

Panos, A. and Gómez, J. (2012). Dealing with Vagueness in Semantic Business Process Management through Fuzzy Ontologies. In *International Workshop on Semantic Business Process Management*.

Solis, J., Pacheco, H., Najera, K., and Estrada, H. (2013). A MDE Framework for Semi-automatic Development of Web Applications. In *1st International Conference on MDE and Software Development*, pages 241–246.

Uschold, M. (2008). Ontology-driven information systems: Past, present and future. In *FOIS'08*, pages 3–18.

van der Aalst, W., Hofstede, A. H. M. T., and Weske, M. (2003). Business process management: A survey. In *1st International Conference on Business Process Management, volume 2678 of LNCS*, pages 1–12.

Wetzstein, B., Ma, Z., Filipowska, A., Kaczmarek, M., Bhiri, S., Losada, S., Lopez-Cob, J.-M., and Cicurel, L. (2007). Semantic business process management: A lifecycle based requirements analysis. In *Workshop on Semantic Business Process and Product Lifecycle Management*.