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
{*ebenezer.sanchez, karen.najera, hugo.estrada, javier.solis*}*@infotec.com.mx*

http://www.infotec.com.mx

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, currently there are still some issues that a BPMS has to cope with, such as the low degree of automation of the BPM life-cycle and the gap between the business and IT views on business processes. In this paper, we present SWB Process, a BPMS driven by Semantic Technologies that provides a higher degree of automation and a better support for modeling, implementation, execution, and analysis phases of BPM life-cycle. We focused on describing how the Ontology-Driven Development approach was used to develop SWB Process and we briefly mention how Semantic Technologies, as basis of SWB Process, support the BPM life-cycle. Our BPMS 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 for improving the efficiency and optimization of enterprise resources and core activities. BPM encompasses a set of methods, techniques and Information Technologies (IT) to manage business processes involving humans, organizations, applications, documents and other sources of information (van der Aalst et al., 2003). BPM is directed by a life-cycle comprising 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). This means that a BPMS supports the modeling of business processes and provides mechanisms to translate those models into an executable system description that helps process performers to accomplish their business activities. Several BPMS has 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), currently there are still some BPM issues to be addressed, such as the low degree of automation of the BPM life-cycle and the gap between the business and IT views on business processes. In order to overcome these problems, researches 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 theoretical and conceptual approaches, that is, BPM ontologies and formalizations (Panos et al., 2012; Oro and Ruffolo, 2012; Mueller, 2012), while others involve prototypes and tools for partially cover the BPM life-cycle, or architectures and functional requirements for a Semantic BPMS (Wetzstein et al., 2007; Francescomarino et al., 2009; Karastoyanova et al., 2008; Stein et al., 2009; Domingue et al., 2013). Nevertheless, there are no works that have addressed the implementation of fully functional and industrial BPMS completely driven by Semantic Technologies. In this paper, we present our proposal which involves a BPMS completely driven by Semantic Technologies (OWL and RDF (OMG, 2004a; OMG, 2004b))

called SWB Process[1]. SWB Process is an industrial and Open Source Semantic BPMS that supports the whole BPM life-cycle. It has been developed following the Ontology-Driven Information Systems approach (Guarino, 1998; Uschold, 2008). Accordingly, ontologies were directly involved in the development of SWB Process through Ontology-Driven Development (Uschold, 2008; Happel and Seedorf, 2006), 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 to increase the degree of automation and to better support modeling, implementation, execution, and analysis phases of BPM life-cycle, thus, closing the gap between business and IT views on business processes.
SWB Process has been successfully validated through real projects, supporting the business processes of several government agencies in Mexico.
Along the paper, we focused on describing how the Ontology-Driven Development approach was used to develop SWB Process and we briefly mention how Semantic Technologies support the BPM life-cycle.
The paper is structured as follows: Section 2 describes the development of SWB Process through the Ontology-Driven Development approach and its main components. Section 3 presents how SWB Process supports BPM life-cycle along with a running example. Section 4 describes the major advantages 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. As mentioned before, Ontology-Driven Information Systems ideas (Guarino, 1998; Uschold, 2008) were followed to define our proposed Semantic BPMS. To do this, the development of SWB Process has been addressed by using SemanticWebBuilder (SWB) (Solis et al., 2013), an Ontology-Driven Development Framework which provides a development methodology and a software platform specialized in Semantic Web application development. Before presenting the development process of SWB Process, we briefly describe SWB in the following subsection.

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

2.1 Development methodology

SWB is an agile Framework for Web application development, where system requirements are modeled into ontologies and from this knowledge representation, the base source code of the new system is automatically generated. As part of the framework, SWB provides three components: 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 from the SWBOntology to Java source code (SWBCodeGen); and c) a platform with several libraries and utilities to accelerate software development and encourage software reuse (SWBPlatform).
The development methodology to generate a new system through SWB implies three major steps: 1) *Modeling*, it comprises the modeling of an ontology with functional requirements of the new system by extending the SWBOntology; 2) *Automatic code generation*, it comprises the execution of the SWBCodeGen to automatically generate the base source code of the new system; and 3) *Specific development*, it comprises the development of the business logic and components to cover each functionality of the new system.
SWB Process has been developed according to the development methodology of SWB, as shown in Fig. 1. A brief description of each step is presented below:

1. *Modeling*: In this step, the SWB Process ontology modeling was performed. This was done by extending the SWBOntology with concepts of the Business Process Model and Notation (BPMN) 2.0 specification (OMG, 2011) (following model to model transformations), as well as concepts coming from our previous BPMS requirements analysis needed for covering BPM life-cycle, such as: user interaction and process execution descriptions.

2. *Automatic code generation*: In this step, the automatic generation of SWB Process base source code was performed following model to code transformations. This was done by executing the SWBCodeGen taking as input the SWB Process ontology. As output the SWBCodeGen provided the SWBP API, which comprises the source code in Java programing language, necessary for persistance of the SWB Process objects.

3. *Specific development*: In this step, components and the operational business logic to implement the SWB Process functionalities were developed. This was done by using the SWB Process base source code encapsulated in the SWBP API and

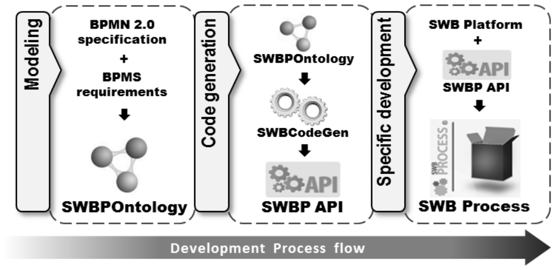

Figure 1: SWB development methodology

the SWBPlatform which provides libraries for connecting to several triplestores for RDF storage.

These steps are described in more detail in the following subsections.

2.2 Modeling the SWB Process Ontology

The SWB Process ontology (SWBPOntology) is an ontology that captures the Business Process Model and Notation (BPMN) 2.0 specification (OMG, 2011), as well as the definition of other concepts coming from our previous BPMS requirements analysis needed for covering BPM life-cycle, such as: services, documents, templates, user interaction components and process execution. Additionally, the ontology considers behavioral aspects of the concepts that it defines.

The modeling of SWBPOntology started with the extension of the SWBOntology provided by the SWB Framework (Solis et al., 2013). This was done for two important reasons: first, the SWBOntology defines classes and properties to support the automatic transformation from the ontology to object-oriented source code in Java language through the SWBCodeGen. Second, SWB Process performs process execution in a Web architecture, and the SWBOntology defines concepts for Web components and Web site features development, which we wanted to reuse.

Then, a manual mapping process between the BPMN 2.0 specification and OWL language was performed. The mapping process started with the selection of classes and properties to be mapped from the BPMN specification, taking into account BPMN class diagrams and text descriptions. The selection was carried out according to the Ontology-Driven Development point of view, that is, considering those elements of BPMN useful for covering all phases of the BPM life-cycle by using ontologies, but not considering elements such as those related to the storage of BPMN diagrams with XML, since SWB Process drives the storage through ontologies with OWL and RDF languages. The selected classes and properties of BPMN were mapped into OWL definitions according the following transformation rules:

- *Rule 1*: A class from the BPMN specification was mapped as a class in the SWBPOntology.
- *Rule 2*: A primitive type property from the BPMN specification was mapped as a datatype property in the SWBPOntology. The domain of the datatype property corresponds to the owner class of the property, whilst the range is an xsd type equivalent to the primitive type.
- *Rule 3*: A complex type property from the BPMN specification was mapped as an object property in the SWBPOntology. The domain and range of the object property correspond to OWL classes related to the complex type property, which were defined in Rule 1 or that are part of the SWBOntology.

After the BPMN to OWL mapping process, concepts derived from our previous BPMS requirements analysis needed for covering BPM life-cycle were defined into SWBPOntology, for instance, 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).

Finally, additional classes were defined to capture behavioral aspects of those concepts already included in the SWBPOntology. Examples of behavioral aspects are: the capacity to catch or send a message and the execution of code blocks (for BPMN script tasks).

It is worth mentioning that the BPMN specification lacks of implementation details, particularly for a business execution engine or the specific implementation of human tasks. At this point, the SWBOntology played a fundamental role, since it includes definitions useful to fill the gaps 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. Moreover, the SWBOntology also provides a mechanism to deal with the multiple inheritance not supported by Java, which takes place in some classes of the BPMN hierarchy. The mechanism comprises the definition of additional classes to control behavioral aspects.

A fragment of the resulting SWBPOntology taxonomy is presented in Fig. 2. The main classes are: the ProcessElement class which is useful to support the BPM modeling phase, the BaseElement, the DataTypes and the ProcessService classes which are useful to support the BPM implementation phase and finally, the Instance class which is useful to support the BPM execution phase. These classes are described below.

ProcessElement. This is the superclass of those classes that define BPMN elements involved in a business process model. This includes the BPMN graphical element definitions such as flow nodes, lanes and pools, as well as process tool definitions such as file templates, time periods and process rules.

BaseElement. This is the superclass of those classes that define BPMN elements that can be reused several times across business process models. For example, classes to manage process documentation or database connections for service configuration.

DataTypes. This is the superclass of those classes that define primitive types used to configure BPMN data objects in a business process model (during the configuration phase of BPM). For example, strings, dates, numbers and URLs.

ProcessService. This is the superclass of those classes that define reusable service definitions, needed for the configuration of Service type tasks. For example, services to send a mail, store data objects in a repository and execute a SPARQL query.

Instance. This is the superclass of those classes defined for business process execution. For example, definitions to control process instance execution and status management of flow nodes in each process instance execution.

2.3 Automatic code generation

The automatic code generation to obtain the base source code of SWB Process was performed by executing the SWBCodeGen. Therefore, it was configured to use the SWBOntology and the SWBPOntology as input for code generation. The SWBCodeGen is a software package that allows to transform ontology definitions (extending from the SWBOntology), into the base source code of a new system. After running the SWBCodeGen, the generated base source code corresponds to a domain-specific and high-level Java API (SWBP API) that encapsulates the source code (classes and methods) necessary to achieve the persistence of the objects involved in the different components of SWB Process. Specifically, the SWBP API includes: 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 source code encapsulated in the SWBP API helps developers to reduce the complexity of managing RDF persistance in a standardized way in which data persistance mechanisms are separated from the business logic of the end application. Thus, accelerating the application development.

2.4 Specific development

Finally, specific development was performed to obtain SWB Process. It consisted in the implementation of components and the operational business logic by using the SWBP API to cover the SWB Process functionalities. The SWBP API is supported by the SWB-Platform which provides a set of libraries that allows to use connectors to several triplestores for RDF storage. In this way, developers can choose from several RDF persistance engines, such as Apache Jena, to achieve RDF persistance without affecting the business logic of the application.

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 3 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 process modeler is a Web based component that supports the BPMN diagram modeling. That is, it allows end users to graphically design processes by using BPMN 2.0 notation. It also performs the validation of the execution semantics of a BPMN diagram according to the BPMN specification, for instance, when to start or end a process and which BPMN elements can be connected in a process flow. In addition, it includes a mechanism to relate ontologies to the BPMN process diagram to define the structure of business artifacts, such as data, documents or Web Services. On the other hand, the SWBP Modeler maps in a transparent manner to the user, each graphical element of a particular BPMN diagram 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* generated by the SWBP Modeler for its configuration in order to make it executable and deployable on a Web site. The configuration comprises two parts: execution configuration and deployment

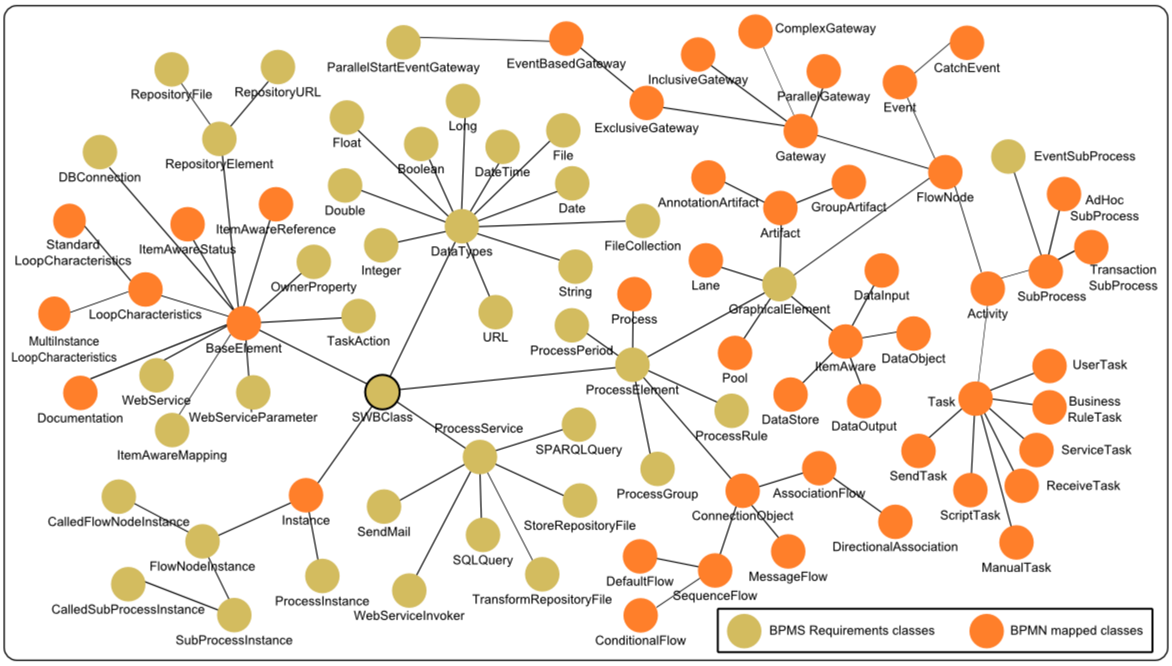

Figure 2: Fragment of the SWB Process Ontology

configuration. The execution configuration consists of capturing the values of each property of the graphical elements comprised in 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. The deployment configuration 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 SWBP ontology 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, for instance, if the task is a service task, the SWBP Engine calls to the corresponding Web Service. Moreover, the SWBP Engine manages the persistance 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, the how Semantic Technologies, as basis of SWB Process, support the BPM life-cycle is described. To do this, for each phase, first, we present the definition proposed in (Wetzstein et al., 2007). Then, we explain how SWB Process gives support to the phase.

- *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*.

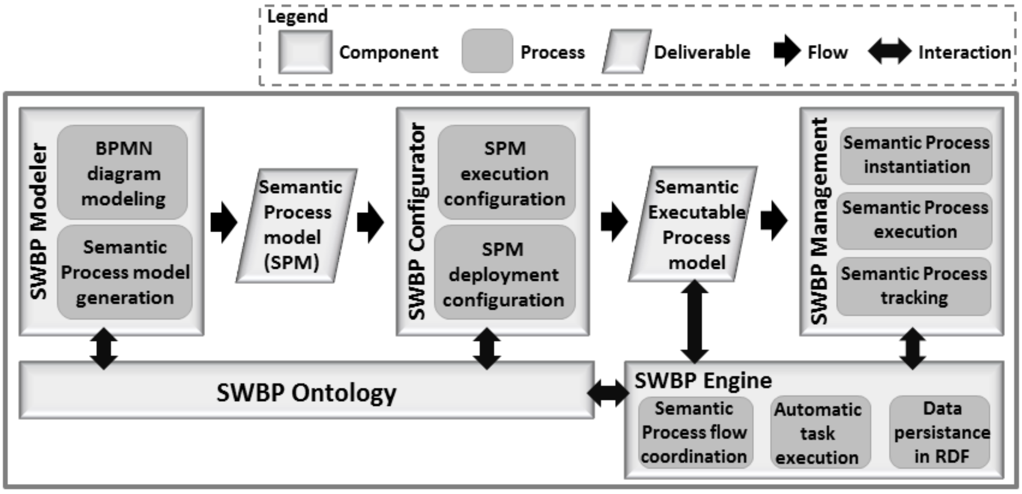

Figure 3: SWB Process Architecture

- *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 *Semantic 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 using SWB Process is presented, which consists of seven steps that cover the BPM life-cycle. The first five steps must be performed by the user 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. The steps are listed below:

Modeling

1. All business artifacts such as documents and data are defined in ontologies by using an Ontology editor.
2. The BPMN diagram (business process flow and business rules) are graphically modeled using the SWBP Modeler.

Implementation

3. Business artifacts (represented as ontologies in step 1) are related with process data objects and activities.
4. Execution properties are configured for each process element.
5. Web form templates of process activities are defined for user interaction using the available properties of each configured data object.

   These steps are performed through the SWBP Configurator. After step 5, the process is ready to be deployed in a wrapper Web page. The SWBP Engine provides process orchestration and data persistence.

Execution

6. Process participants can create a process instance and perform their process tasks.

Analysis

7. Monitoring, tracking and querying process data can be performed through the SWBP Management.

Following, an example of a process generated with SWB Process is described. The example corresponds to the abstract of a process 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.

The participants involved in the process 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 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. 4 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

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

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

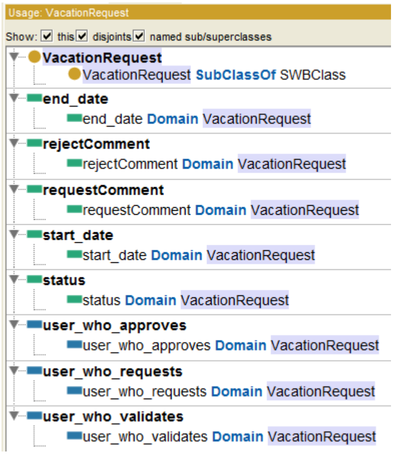

Figure 4: Vacation request data modeled in Protege

rules by using the *SWBP modeler*. Fig 5 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. 6, 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 definition. 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. 7 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

Our approach corresponds to a Semantic BPMS, whose development was driven by the SWBPOntology, the same that drives the BPM life-cycle. The benefits of using an ontology for the development of SWB Process are: a) System requirements are

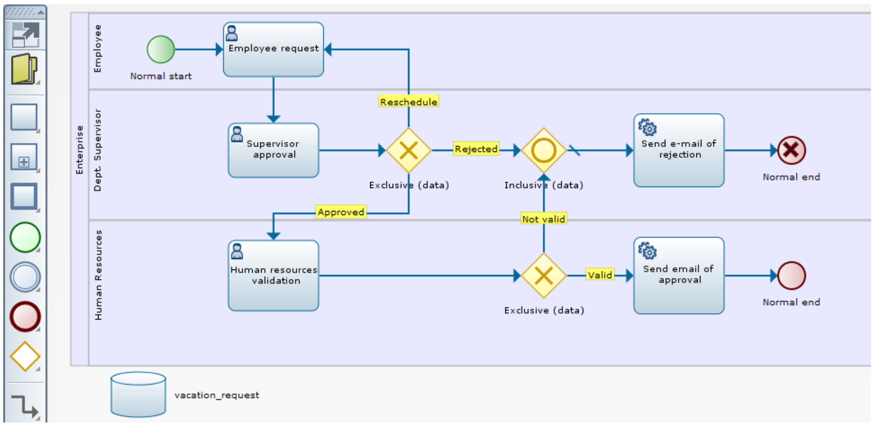

Figure 5: Vacation request process modeled with SWB Process modeler

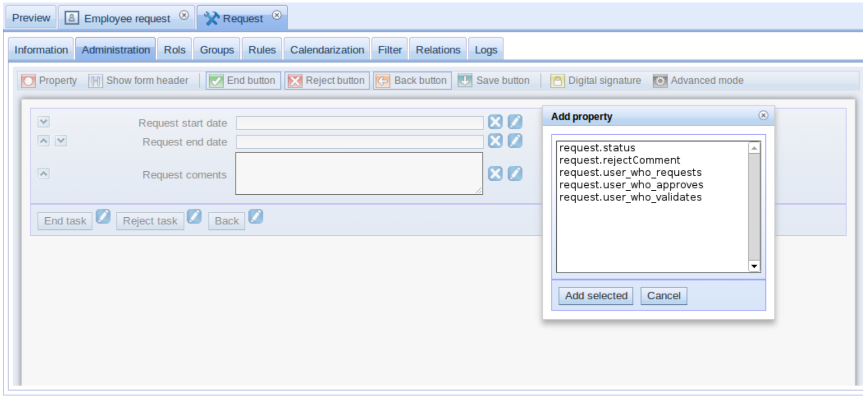

Figure 6: Employee request task configuration

captured into the SWBPOntology which allows automatic generation of source code through model to code transformations; b) the source code could be automatically re-generated from the SWBPOntology if system requirements change, making SWB Process flexible to be adapted to new BPM needs, thus improving maintenance and update tasks; c) the SWBPOntology represents the architecture, behavior and data schema of SWB Process, therefore, if requirements change only a single model (the SWBPOntology) has to be modified to support the changes needed in SWB Process; and d) the system architecture is described with Semantic Web standards, allowing interoperability with other components or IT systems.

The benefits of using ontologies at run time, when SWB Process is applied to the BPM life-cycle in organizations, are: a) SWB Process provides a mechanism to make explicit the knowledge involved in a business process by means of modeling business artifacts into ontologies; b) ontologies provide a machine readable representation of business knowledge involved in the process itself and in the process orchestration, enabling automatic process space querying and reasoning tasks; c) the business process knowledge is described with Semantic Web standards, allowing interoperability with other business processes; and finally, d) other Semantic Web paradigms such as Linked Data can be applied to exploit business knowledge.

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 proper-

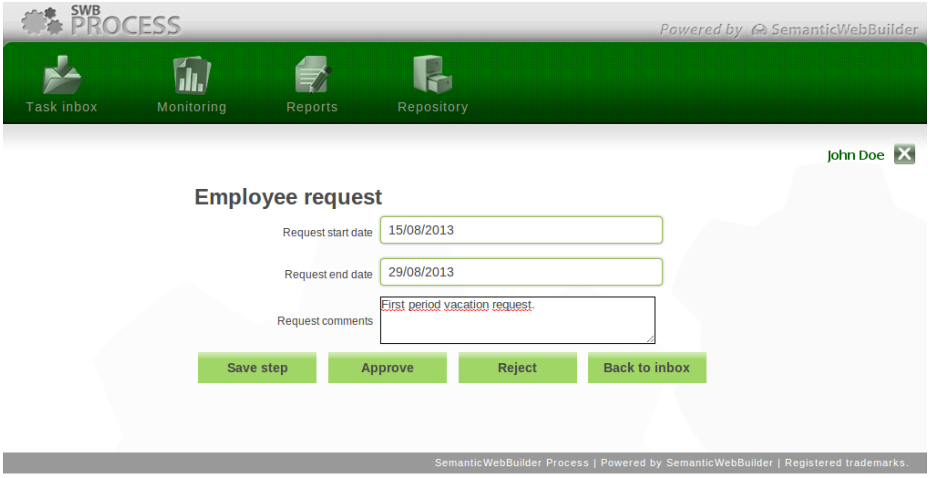

Figure 7: Employee request task deployment

ties of the BPMN 2.0 and the Service Component Architecture (SCA) assembly model. Some applications of this ontologies in the BPM life-cycle are described. In the work of Alexopoulos (Panos et al., 2012), authors argue that the use of fuzzy ontologies is useful for the effective capture, representation and exploitation of knowledge that is vague in a business process.

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, authors propounded a Semantic Execution Environment and show how existing BPMS components can be extended with semantic features. In works of Stein (Stein et al., 2009) an extension to the BPM ARIS tool is presented. It comprises the annotation of EPC process models with concepts of the WSMO ontology, enabling WS discovery during process execution. 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.

In our approach, we present a Semantic BPMS that has been developed by using Semantic Technologies. Thus, an ontology plays the role of data architecture and processes data is persisted in RDF. On this semantic basis, ontologies drive the whole BPM life-cycle. At modeling, process flow, process data and related artifacts are represented as ontological definitions, making explicit the implicit business knowledge. At implementation and execution, a process instance is represented as ontology individuals in which querying and automatic reasoning can be applied at analysis phase. Furthermore, our proposal is easily extendable to support new BPM needs.

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 to increase the degree of automation and to better support the BPM life-cycle, since SWB Process allows business processes and business artifacts to be quickly (re)modeled, (re)implemented, and (re)executed according to the continuous changes in organizations, thus, closing the gap between business and IT views on business process.

The use of Semantic Technology with SWB Process provides the following advantages: a) a BPMS with flexible and agile mechanisms to adapt to new BPM needs, reducing code maintenance issues and increasing reuse of code, b) explicit meaning to the information implicitly represented in a business model, c) ma-

chines, as well as people are enabled to understand, share and reason over business processes models and information, d) 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).

SWB Process is an industrial project which is distributed as an Open Source system (it can be downloaded and used free of charge) with the purpose of providing a tool to support business process implementation in government and enterprises. In tandem with SWB Process, we provide the following support services: consultancy, mentoring, technical support, training and customization.

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.

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

Domingue, J. et al. (2013). SUPER. Semantics Utilised for Process Management within and between Enterprises. 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.

Francescomarino, C. D., Ghidini, C., Rospocher, M., Serafini, L., and Tonella, P. (2009). Semantically-aided business process modeling. volume 5823 of *Lecture Notes in Computer Science*, pages 114–129. Springer.

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

Happel, H.-J. and Seedorf, S. (2006). Applications of ontologies in software engineering. In *SWESE'06*, Athens, USA.

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, 2005. ICEBE 2005. IEEE International Conference on*, 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*, page 371372.

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/. Last access: 05/07/2013.

OMG (2004b). Resource Description Framework (RDF). Online. http://www.w3.org/RDF/. Last access: 05/07/2013.

OMG (2011). Business Process Model and Notation (BPMN) 2.0. http://www.omg.org/spec/BPMN/2.0/. Last access: 05/07/2013.

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

Panos, A., Gómez-Pérez, and José-Manuel (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 *Proceedings of the 1st International Conference on MDE and Software Development*, pages 241–246.

Stein, S., Stamber, C., and Kharbili, M. (2009). Aris for semantic business process management. In Ardagna, D., Mecella, M., and Yang, J., editors, *Business Process Management Workshops*, volume 17 of *Lecture Notes in Business Information Processing*, pages 498–509.

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 *Proceedings of the 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*.

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