Proceedings of the Federated Conference on
Computer Science and Information Systems pp. 1307–1314

DOI: 10.15439/2015F250
ACSIS, Vol. 5

Opportunities for Business Process Semantization in Open-source Process Execution Environments

Krzysztof Kluza and Krzysztof Kaczor and Grzegorz J. Nalepa and Mateusz Ślażyński
AGH University of Science and Technology
al. A. Mickiewicza 30, 30-059 Krakow, Poland
E-mail: {kluza,kk,gjn,mslaz}@agh.edu.pl

***Abstract*—Business Process models help to visualize the processes of an organization. There are open-source process execution engines that provide the environments for enacting process models. However, they lack semantization capabilities. In this paper, an overview of Business Process semantization techniques is provided. Moreover, we discuss the common architecture of the selected open-source process execution environments (Activiti, jBPM and Camunda) and provide the insights how they can be improved using semantization methods. We also present the use of the introduced techniques in the Prosecco (Processes Semantics Collaboration for Companies) research project.**

***Index Terms*—Business Process Model and Notation (BPMN), Business Process Semantization, Business Process Management**

I. Introduction

BUSINESS Process Management (BPM) [1] is a modern approach to improve organization's workflow, which focuses on reengineering of processes in order to optimize procedures, increase efficiency and effectiveness by constant process improvement. A Business Process (BP) model constitute a graphical representation of a process in an organization, which is composed of related tasks that produce a specific service or product for a particular customer [2].

The BP runtime environment manages and monitors processes as they perform. It orchestrates the activities and interactions of the process with web services or other third-party applications. Such software also supports user tasks, handles exceptions or escalations by tracing the workflow in the model. There are more than 70 BPMN implementers[1],

As process models can be ambiguous, the top leader vendors that develop proprietary software introduced semantization techniques to their solutions. This helps to support more intelligent functions, like web services discovery or element name suggestions. These techniques use semantic annotations, which can be based on a formally specified ontology.

For the purpose of this paper, we focused on the selected open-source process execution environments that are freely accessible, use the BPMN 2.0 notation [3] on the executable level and can be easily extended with new functions. We analyze the possible ways of enhancing them with semantization methods. Some of these methods have been introduced in the Prosecco (Processes Semantics Collaboration for Companies) research project[2], which takes advantage of the Activiti execution engine.

The paper is structured as follows. In Section II, we present selected open-source process execution environments and their general architecture. Section III gives an overview of business process semantization approaches, especially focusing on the solution developed in the SUPER project and the SAP AG company. Section IV analyzes the possibilities of process semantization in the open-source process execution environments. The paper is concluded in Section V.

II. Open-source Process Execution Environments

All three environments, Activiti[3] [4], jBPM[4] [5], and Camunda[5] [6], are light-weight BPM suites with extensible BPMN 2.0 process engines. They can be run in any Java environment, embedded in an application or as a service. The engines allow a user to execute business processes using the BPMN 2.0 specification, and are open-source software distributed under the Apache license.

All the projects include such components as:

- Process Engine – an execution engine that provides Process Virtual Machine; the engine uses BPMN 2.0 as the underlying XML format for the process definitions.
- Web-based Process Modeler – a web-based process editor for modeling business processes (mainly for analysts).
- Eclipse-based Process Designer – more complex process editor for modeling and implementing detail aspects of process models, it allows graphical modeling, development and debugging of models.
- Process Management Interface – a web application providing access to the runtime engine for all users of the system, this includes task and job management, process instance inspection, viewing reports based on statistical history data, etc.
- Process Repository – a repository for storing and managing process definitions.
- History Log – a log for storing history information about the process instances that are being executed, which can be further used to generate reports, etc.

The overview of this common architecture is presented in Figure 1 and comparison of their components is provided in Table I. Such an architecture is further considered for semantization of the execution environment.

[1] See: http://www.bpmn.org/#tabs-implementers.

[2] See: http://prosecco.agh.edu.pl.

[3] See: http://www.activiti.org/userguide/.

[4] See: http://www.jbpm.org/learn/documentation.html.

[5] See: http://docs.camunda.org/guides/user-guide/.

978-83-60810-66-8/$25.00©2015, IEEE

L.p.	Component	Activiti	jBPM	Camunda
1.	**Process Engine**	Activiti Engine	Core Process Engine	Camunda Process Engine
2.	**Process Repository**	part of Activiti Engine	Process Repository	part of Camunda Process Engine
3.	**Web-based Process Modeler**	Activiti Modeler	Web-based Designer	Camunda-bpmn.js
4.	**Eclipse-based Process Designer**	Activiti Designer	Eclipse-based Editor	Camunda Modeler
5.	**Process Management Interface**	Activiti Explorer	jBPM Console	Camunda Cockpit/Tasklist
6.	**History Log**	part of Activiti Engine	History Log	part of Camunda Process Engine

Table I
COMPARISON OF ACTIVITI, JBPM AND CAMUNDA COMPONENTS

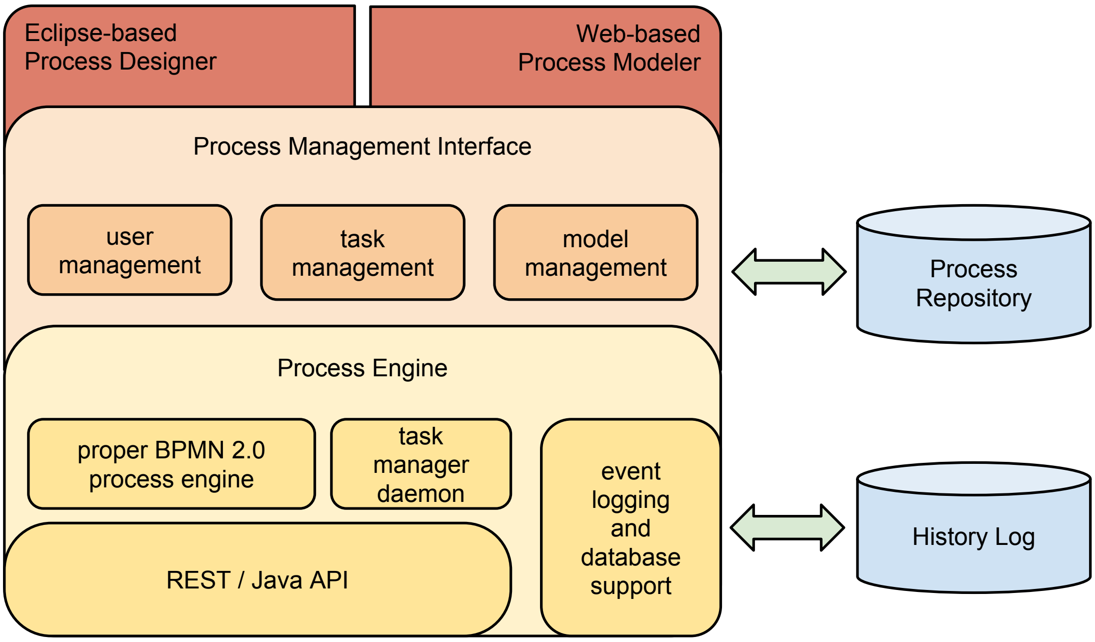
Eclipse-based Process Designer
Web-based Process Modeler
Process Management Interface
user management
task management
model management
Process Repository
Process Engine
proper BPMN 2.0 process engine
task manager daemon
event logging and database support
REST / Java API
History Log

Figure 1. The general architecture of the analyzed environments (based on our comparison of the Activiti, jBPM and Camunda environments)

III. Overview of Business Process Semantization Possibilities

This section presents the overview of the semantization approaches for business processes and business process runtime environments, based on the analysis of the research results in this field. It focuses on Business Process semantization in: the SUPER project, the BPM product of the SAP AG company, as well as some substitute of semantization which can be observed in Signavio Process Editor.

A. Business Process Semantization in the SUPER project

The goal of the SUPER project [6] was to create tools for business process semantization by describing process models using concepts from the ontology.

A BPM system that uses ontology as a common language of communication can facilitate clear expressing the statement by the people from the business and provide a method of unambiguous communication between the system, IT people and non-technical people associated with the business.

The SUPER project architecture

The elements of the architecture of the SUPER environment is presented in Figure 2. WSMO Studio is a stand-alone application (also available as a plugin for the Eclipse integrated development environment), which provides the following functions: enables customers to create ontologies, specify goals, web services and mediators, as well as provide appropriate interfaces for these elements. Additionally, the environment provides dedicated editors, including SAWSDL editor to annotate semantics to WSDL.

Ontologies in the SUPER project

In the SUPER project ontology the following elements can be distinguished (the relationship between them is presented in Figure 3):

- Web Service Modeling Ontology [7] specifies formally the terminology of the information used by all other components and provides the semantic description of web services (their functional and non-functional properties, and their interfaces).
- Business Domain Ontologies related to the business domain knowledge (Business Functions, Business Process Resources, Business Roles and Business Modeling Guidelines Ontologies).

[6] The website of the SUPER project http://www.ip-super.org/content/view/196/163/ is no longer maintained. However, some pieces of information about the results can be found at: http://www.sti-innsbruck.at/results/movies/sbp-execution-developed-in-super as well as in the project publications.

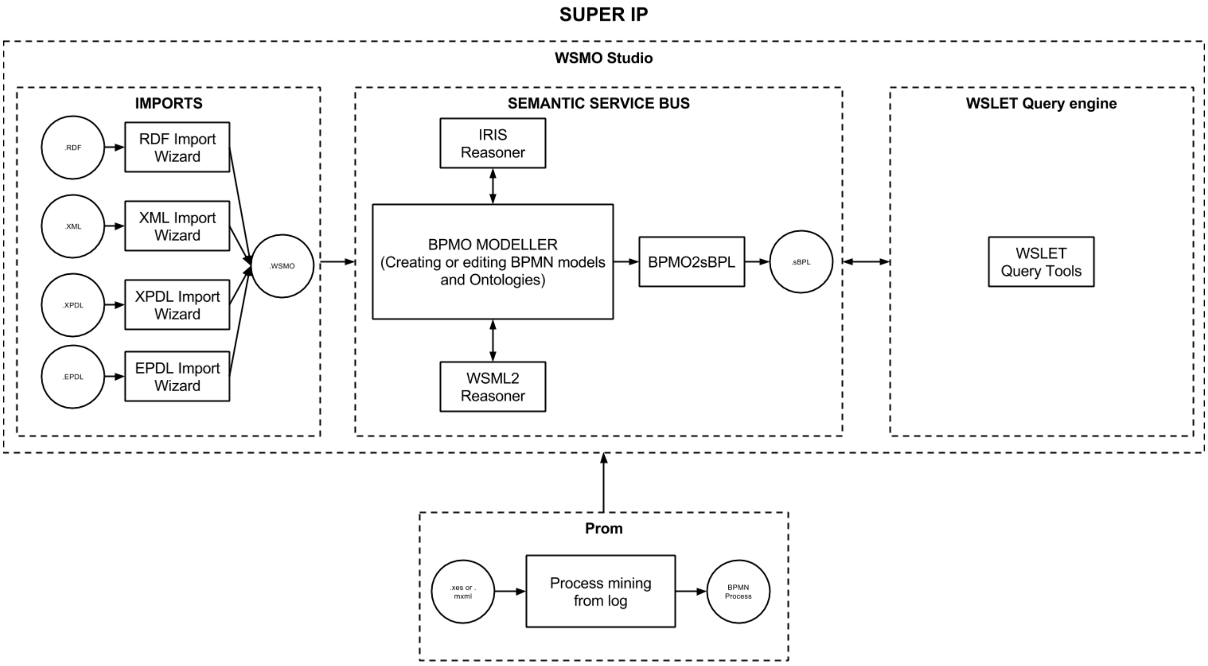

Figure 2. The elements of the basic architecture of the SUPER project (based on the description from the SUPER project website)

- SUPER Ontology Stack:
  - Upper-Level Process Ontology (UPO) is the top-level ontology that aims to represent high-level concepts for Business Process modelling.
  - Business Process Modelling Ontology (BPMO) acts as a bridge between the business level and the processes execution level and is used for representing high-level business process workflows.
  - Semantic Event-driven Process Chains notation Ontology (sEPC) supports the annotation of process models created with EPC tools.
  - Semantic Business Process Modeling Notation Ontology (sBPMN) for formalization of the core subset of the BPMN notation.
  - Semantic BPEL Ontology (sBPEL) extends the ontology of BPEL with a Semantic Web Services model.
  - Behavioral Reasoning Ontology (BRO) for reasoning over the business processes behaviours using WSML axioms.
  - Events Ontology (EVO) that constitutes a reference model for capturing logging information used by the execution engines and the analysis tools.

The SUPER project process life cycle

The methodology of the SUPER project defines four phases that form the business process life cycle [9]. For these phases the appropriate methods and techniques for business process semantization were developed. In the following paragraphs, these phases are elaborated.

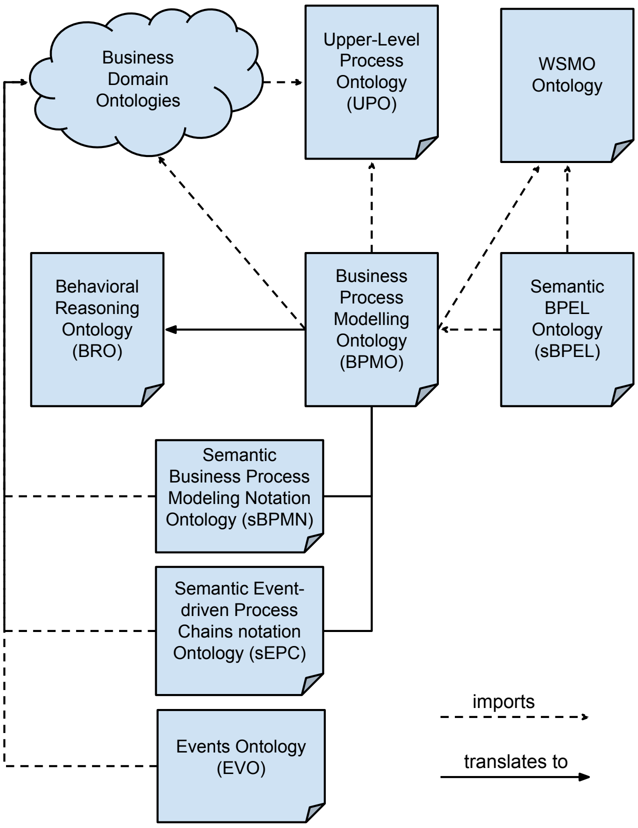
Business Domain Ontologies
Upper-Level Process Ontology (UPO)
WSMO Ontology
Behavioral Reasoning Ontology (BRO)
Business Process Modelling Ontology (BPMO)
Semantic BPEL Ontology (sBPEL)
Semantic Business Process Modeling Notation Ontology (sBPMN)
Semantic Event-driven Process Chains notation Ontology (sEPC)
Events Ontology (EVO)
imports
translates to

Figure 3. The overview of the ontologies in the SUPER project (based on [8])

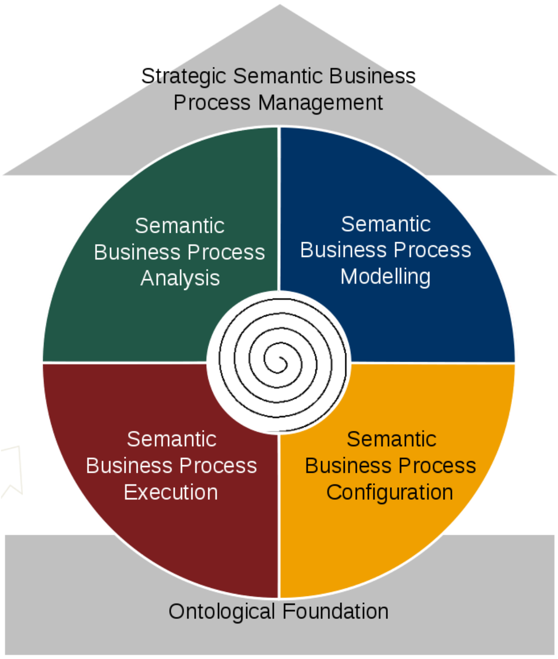
Strategic Semantic Business Process Management
Semantic Business Process Analysis
Semantic Business Process Modelling
Semantic Business Process Execution
Semantic Business Process Configuration
Ontological Foundation

Figure 4. Phases in the SUPER project methodology [9]

Phase 1: Semantic Business Process Modelling

The first phase of the life cycle in the SUPER methodology consists in developing business process models based on the BPMO ontology. It uses the environment for semantic modeling (WSMO Studio tools with the integrated BPMO editor). The business process model is based on the domain ontologies specified for the particular company as well as on Semantic Web Services and Goals. Its source can be implicit knowledge of business analysts or analysis of reports from the previous Semantic Business Process (SBP) Analysis phase.

Phase 2: Semantic Business Process Configuration

Semantic Business Process Configuration (SBPC) is a second phase in the SUPER methodology life cycle that uses the semantic business process models which are the output from the previous phase. During this phase, the semantic business process models are configured.

The configuration phase consists of deriving an sBPEL ontology from a BPMO instance, discovering the possible Semantic Web Services (SWS) [10], identifying the potential data mismatches, and based on them creating the interface mappings and data mediators. Lastly, the process is validated in terms of the correctness of the semantic process description before the execution and potentially refined.

Phase 3: Semantic Business Process Execution

In the third phase of SUPER methodology, modeled and configured processes are executed and processed. During the runtime, data, which will be used for analysis, are collected. As this phase is performed without user interactions, it minimizes the time required for its completion. In this pahse, process execution is supported by the semantic BPEL (BPEL4SWS) and detection and execution of Semantic Web Services (SWS).

In Figure 5, the scenario of semantic business process execution is presented. This scenario involves the following seven steps [9]:

1) *Request Service* – in order to initialize a semantic BPEL process, a user have to send request through the Semantic Service Bus to SBPELEE.
2) *Achieve Goal* – invocation of SWS is delegated to SEE by SBPELEE which passes the WSMO Goal to it.
3) *Discover Service* – SEE queries the Semantic Web Services repository to discover the desired SWS.
4) *Invoke Service* – SEE invokes the discovered SWS.
5) *Engine Return Result* – SEE returns the result received from SWS to SBPELEE.
6) *User Return Result* – After the process execution has been finished, the result is returned to the user.
7) *Process Tracking* – During the execution, execution events are published to Execution History for persistence and to the Monitoring Tool for tracking process executions.

The most important benefits of using such an approach are:

- flexible use of Web Services,
- supplier matching supported by Semantic Web Service discovery and invocation from within semantic business processes,
- more flexible traffic routing,
- automates supplier matching and traffic routing process taking into account all existing suppliers,
- minimizes time-to-offer.

Phase 4: Semantic Business Process Analysis

The last phase of the process life cycle concerns the analysis of the executed processes. In this phase, various analysis goals are supported, such as: overview over process usage, detecting business and technical exceptions, etc.

Thanks to this phase, it is possible to get an overall overview about system usage, finding out exceptions within process flow and bottlenecks, as well as get necessary information needed to apply 6 Sigma methodology.

Data, Information, and Process Integration with SWS

The SUPER project took advantage of the experience of the DIP[7] (Data, Information, and Process Integration with Semantic Web Services) [11] platform.

The aim of the DIP platform was extending the semantic web technologies and web services in order to create a new technical infrastructure – Semantic Web Services (SWS).

The DIP platform provides: Web Service Modelling Ontology (WSMO), Web Service Modeling Language (WSML) as the language for modeling web services, and Web Service Execution Environment (WSMX) as software framework for runtime binding of service requesters and service providers.

[7] See: http://dip.semanticweb.org/.

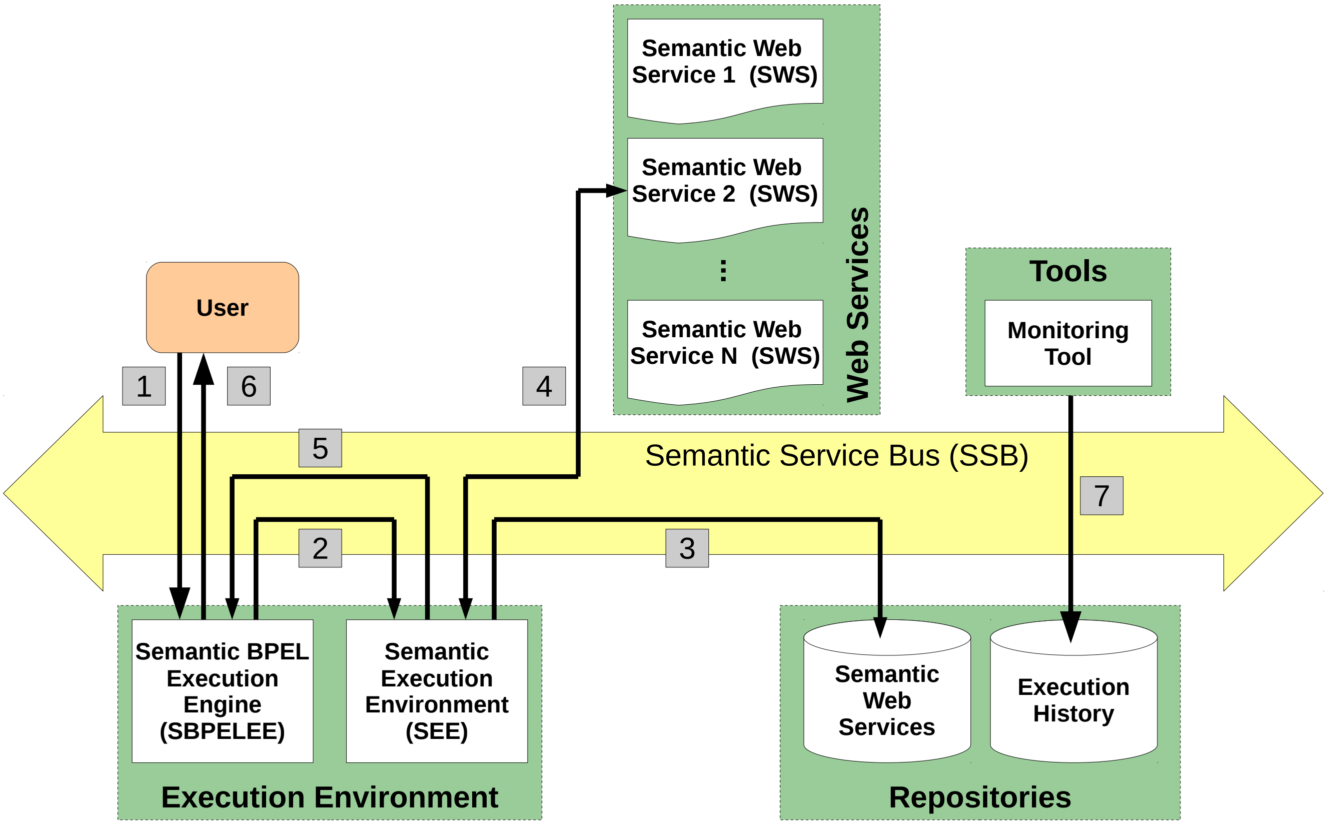
Semantic Web Service 1 (SWS)
Semantic Web Service 2 (SWS)
⋮
Semantic Web Service N (SWS)
Web Services
Tools
Monitoring Tool
User
1
6
4
5
Semantic Service Bus (SSB)
7
2
3
Semantic BPEL Execution Engine (SBPELEE)
Semantic Execution Environment (SEE)
Execution Environment
Semantic Web Services
Execution History
Repositories

Figure 5. The execution scenario in the SUPER methodology (based on the D10.2 SUPER Showcase Presentation http://slideplayer.com/slide/741902/)

B. *Business Process Semantization in SAP AG*

Another more extended solution related to business process semantization is an approach developed [12] and patented[8] by the SAP AG company.

Semantization of business processes in the SAP AG uses the semantic descriptions for business process artifacts. The integration consists in linking the identified semantic pieces of information described in the form of ontology with the elements of business process models.

The approach also supports semantic modeling by matching elements of a process model to concepts from ontologies and using fitting functions for choosing proper semantic annotations. This is achieved by comparing the given context and text description with instance domain ontology. In the approach, three goals are achieved: support for modeling, exploring relevant services, and searching the process model repository.

Figure 6 shows the main components of the process modeling tool semantic extension. The BPMN data objects are used for describing activities by defining the related objects and state transitions. For such activities, a user can graphically specify their preconditions and postconditions, as well as define the related objects with the specification of the object state changes before and after the execution of the activity.

An ontology in this approach contains the information about objects, states, state transitions, and actions related to the domain. For each object the possible states and state transitions are defined and they form the object life cycle. These kinds of domain ontologies support semantic process modeling by using their concepts in model elements specification, especially by suggesting relevant concepts or instances of objects. For suggesting the relevant components (data objects, activities, associations and states), a combination of different algorithms associated with the text matching was used. The algorithms take advantage of contextual information related to the process model as well as domain knowledge ontology.

Thanks to the domain knowledge, the names of tasks can be suggested based on the object life cycle. The object life cycle can also be used to exclude re-using the task names that have already been modeled.

The system also supports the semantic description of data flow. The object status can be visualized directly in the diagram. The “less than” sign (<) denotes the object status before and the “greater than” (>) denotes the object status after performing the associated tasks.

Such semantization supports consistency checking and extends the capabilities of semantic searching. Compared to the approach of the SUPER project, it supports more flexible and accurate semantic annotations by referring directly to the elements from the defined domain ontology.

[8] See: The patent “Semantic extensions of business process modeling tools” number US 8112257 B2: https://www.google.com/patents/US8112257.

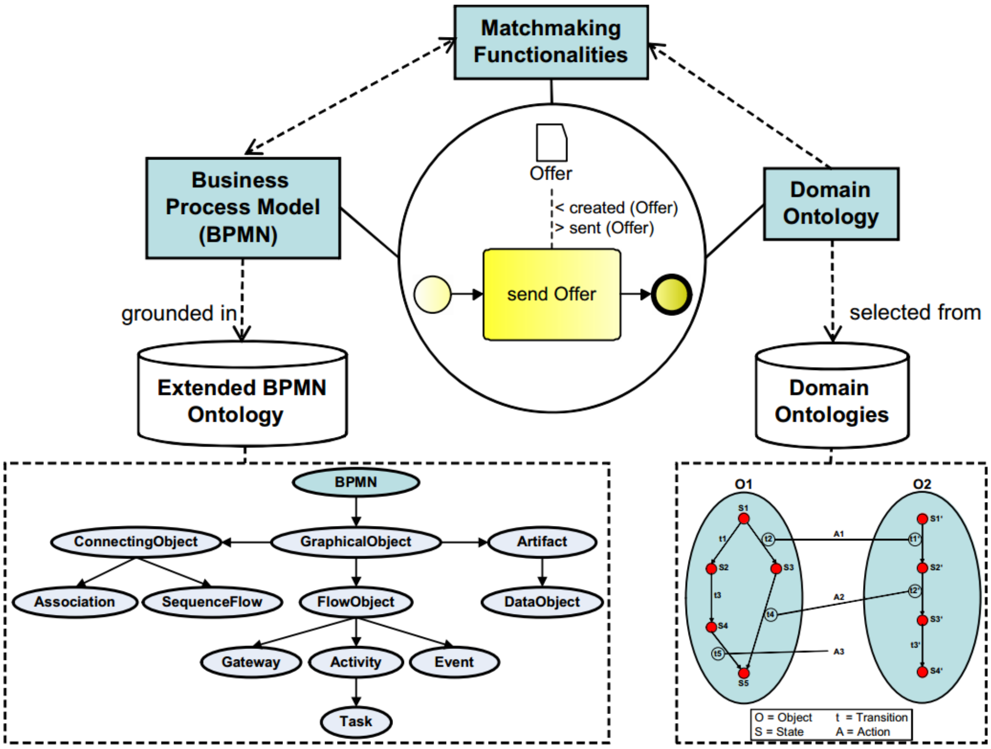

Figure 6. The main components of the semantic extension tools for modeling business processes in the SAP platform [12]

C. Other possibilities of semantization

As the considered open-source process execution environments use REST interfaces, the possibility of semantization of the REST interface is presented in [13]. The authors compare the different semantic annotation languages for REST interfaces and show how to take advantage of them by creating a website which combines online applications from different sources (in particular internet services) - the so-called mashup. They also proposed a new language SAWADL, based on WADL (Web Application Description Language), for REST Web Services semantization.

In the Signavio Process Editor[9], some basic semantization in the form of a dictionary can be observed. In the dictionary one can define the concepts, assign them to one of the 6 categories (Organizational Unit, Documents, Activities, Events, IT System, Other), add the appropriate descriptions, as well as assign to them additional documents (links to them). Then, these concepts can be used to describe the elements of the BPMN model, e.g. during choosing a name for the particular task in the process (see Figure 7).

Although the tool does not support the formal semantic description in the form of the ontology, it supports multilingual description of the same concepts what allows users to work with the same model in different countries in their own languages.

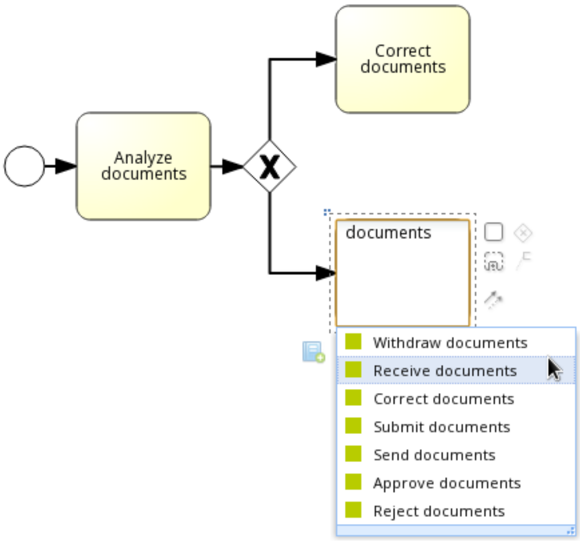

Figure 7. Using dictionary during modeling in the Signavio Process Editor

Other works related to semantic business process modeling can be found in the papers [14], [15], where the processes in the form of Petri nets are connected with the ontology describing the network. Their objective is to standardize the terminology, in particular with regard to the level of abstraction of the labels used in the model. This allows for validating models and detecting the incorrect names of the elements.

[9] See: http://academic.signavio.com.

IV. Analysis of semantization of selected runtime environments

A. Prosecco System

The Prosecco (Processes Semantics Collaboration for Companies) project aims at addressing the needs and constraints of Small and Medium Enterprises (SME) by designing methods that will significantly improve BPM systems by simplification of the system design and configuration, targeting the management quality and competitiveness improvement, fostering decisions making and strategic planning in the SME market sector. The Prosecco system involves three technologies for specification of business logic: business processes, rules and ontologies. Additionally, these technologies are supported by external services providing additional functionalities and integrating core of the system with external tools.

The architecture of the system is oriented towards services what significantly improves its portability and high versatility. Such architecture also enables integration with external tools that provide their functionality as a service. As each element of the architecture provides own data management, there is no centralized repository for all models. Therefore, Prosecco repository consists of several repositories for various models:

1) Repository for Business Processes is called *prosecco-business* and is managed by Activiti engine. It stores information concerning existing processes and their instances, variables and other data processed by the engine. Additionally, the repository contains components that can be used for creating new processes.
2) *Prosecco-knowledgebase* is a repository for rules. It is divided into two parts. The first part stores rules processed by Drools rule engine. The second contains rules learned according to decisions made by the users that are traced on the business process level.
3) Ontology is stored in the OWL format. The POJO model, which is suitable for process and rule engines, can be generated from the OWL representation.
4) *Prosecco-profilemanager* repository stores information related to users and ACL (Access Control List).
5) System History is managed by the Cassandra tool and stores information concerning operations performed within the system.

In turn, due to the fact of usage of the ontology, the project assumes that all data types and their instances existing within the system are consistent with the ontology. Because of the separate data repositories, the object types and existing instances have to be continuously synchronized with the ontology. For example for rule engine the POJO (Plain Old Java Object) model is generated according to the ontology, and for external tools dedicated integration interfaces providing type alignment were developed.

The Prosecco system uses Activiti as the process engine. Apart from the system types generated according to ontology, the Activiti uses also other semantization techniques. The possible scenarios of semantization of the open-source process engines are considered in the following section.

B. Analysis of semantization scenarios of Activiti, jBPM and Camunda runtime environments

In Section II, six components of the open-source Activiti, jBPM and Camunda environments are distinguished. Semantization of each of them allows for disambiguation of data description and controlling their integrity. Moreover, it may extend their functionality and possible use scenarios in the following way:

1) Process Engine – may allow for invoking semantically matched web services, rule-based components or subprocesses.
2) Repository of Processes – may enable semantic search of models in the repository.
3) Web-based editor of Processes – may provide semantic-based recommendation of the elements during the modeling process.
4) Eclipse-based editor of Processes – may suggest names of a process elements, artifacts, etc.
5) Process Management System – may support semantic search in the system e.g. running process instances.
6) Log – logging of system events described semantically.

Semantization of these components can bring advantages. Some of these semantization scenarios are considered in the Prosecco project.

Moreover, Prosecco also extends Activiti with a module supporting additional data types. Activiti provides poor data type system[10]; natively, it supports only five data types (`String`, `Long`, `Enum`, `Date`, `Boolean`) that are often insufficient[11]. Therefore, the developed module provides dedicated form-based user interface that allows for entering the values and provides validation mechanism for such new data types as:

- `Month` and `Season`, `Text`,
- `Double` (floating point number),
- `PIN` (Personal Identification Number),
- `PESEL` (Polish national identification number),
- `Country` (list of countries).

Semantization of this module may include additional descriptions of these types as well as their instances, e.g.:

- `Month` – number of days,
- `Season` – months or days in particular season,
- `Text` – minimal or maximal length of the text,
- `Double` – number of significant digits, unit, displaying format, etc.,
- `PIN` – number of digits, number of attempts to enter correct value, additional security information,
- `PESEL` – interpretation of digits in the number (e.g. birth date, gender),
- `Country` – geographical location, polity, etc.

According to the project assumptions, all the types as well as their semantic descriptions that are used within process must be consistent with the ontology.

[10] See: http://www.activiti.org/userguide/#formProperties.

[11] The Activiti environment provides also two additional advanced data types: `User` and `ProcessDefinition`, which are not listed in the user guide, but available in the environment [4].

V. Summary and future works

The paper gives an overview of business process semantization approaches in the existing proprietary software. It also presents the selected open-source process execution environments (such as Activiti, jBPM and Camunda) and outlines their general architecture, as well as analyzes the possibilities of process semantization in these open-source environments.

Some of the proposed semantization methods have already been introduced in the Prosecco (Processes Semantics Collaboration for Companies) research project, which takes advantage of the Activiti execution engine. Activiti was chosen because it can be easily extended and provides a good documentation [4]. Activiti is the winner of a 2013 Best of Open Source Software Awards (BOSSIE)[12].

Our future works will focus on development of new semantization techniques that can improve business process management environments not only with simple semantic annotations [16]. This can be used for extending recommendation methods in Activiti [17] or semantization of rules in the Wiki environment integrated with processes [18]. Moreover, it is possible to take advantage of semantization in business process verification [19].

References

[1] M. Weske, *Business process management: concepts, languages, architectures*. Springer Science & Business Media, 2012.

[2] A. Lindsay, D. Dawns, and K. Lunn, "Business processes – attempts to find a definition," *Information and Software Technology*, vol. 45, no. 15, pp. 1015–1019, Dec 2003, elsevier.

[3] OMG, "Business Process Model and Notation (BPMN): Version 2.0 specification," Object Management Group, Tech. Rep. formal/2011-01-03, January 2011.

[4] T. Rademakers, T. Baeyens, and J. Barrez, *Activiti in Action: Executable Business Processes in BPMN 2.0*, ser. Manning Pubs Co Series. Manning Publications Company, 2012.

[5] M. Salatino, *jBPM Developer Guide*. Packt Publishing Ltd, 2009.

[6] J. Freund and B. Rücker, *Real-life BPMN: Using BPMN 2.0 to analyze, improve, and automate processes in your company*. Camunda, 2012.

[7] D. Roman, U. Keller, H. Lausen, J. de Bruijn, R. Lara, M. Stollberg, A. Polleres, C. Feier, C. Bussler, D. Fensel *et al.*, "Web service modeling ontology." *Applied ontology*, vol. 1, no. 1, pp. 77–106, 2005.

[8] A. Filipowska, M. Hepp, M. Kaczmarek, and I. Markovic, "Organisational ontology framework for semantic business process management," in *Business information systems*. Springer, 2009, pp. 1–12.

[9] I. Weber, J. Hoffmann, J. Mendling, and J. Nitzsche, "Towards a methodology for semantic business process modeling and configuration," in *Service-Oriented Computing - ICSOC 2007 Workshops*, ser. Lecture Notes in Computer Science, E. Di Nitto and M. Ripeanu, Eds. Springer Berlin Heidelberg, 2009, vol. 4907, pp. 176–187. [Online]. Available: http://dx.doi.org/10.1007/978-3-540-93851-4_18

[10] D. Fensel, F. M. Facca, E. Simperl, and I. Toma, "What are sws good for? dip, super, and soa4all use cases," in *Semantic Web Services*. Springer, 2011, pp. 299–324.

[11] D. Fensel, H. Lausen, A. Polleres, J. de Bruijn, M. Stollberg, D. Roman, and J. Domingue, *Enabling semantic web services: the web service modeling ontology*. Springer Science & Business Media, 2006.

[12] M. Born, F. Dörr, and I. Weber, "User-friendly semantic annotation in business process modeling," in *Web Information Systems Engineering – WISE 2007 Workshops*, ser. Lecture Notes in Computer Science, M. Weske, M.-S. Hacid, and C. Godart, Eds. Springer Berlin Heidelberg, 2007, vol. 4832, pp. 260–271. [Online]. Available: http://dx.doi.org/10.1007/978-3-540-77010-7_25

[13] A. Malki and S. M. Benslimane, "Building semantic mashup," in *ICWIT*, 2012, pp. 40–49.

[14] A. Koschmider and E. Blanchard, "User assistance for business process model decomposition," in *In First IEEE International Conference on Research Challenges in Information Science*, 2007, pp. 445–454.

[15] A. Koschmider and A. Oberweis, "Ontology based business process description." in *EMOI-INTEROP*, 2005.

[16] Y. Liao, M. Lezoche, H. Panetto, N. Boudjlida, and E. R. Loures, "Semantic annotation for knowledge explicitation in a product lifecycle management context: A survey," *Computers in Industry*, vol. 71, pp. 24–34, 2015.

[17] S. Bobek, G. J. Nalepa, and O. Grodzki, "Integration of activity modeller with bayesian network based recommender for business processes," in *Proceedings of 10th Workshop on Knowledge Engineering and Software Engineering (KESE10) co-located with 21st European Conference on Artificial Intelligence (ECAI 2014), Prague, Czech Republic, August 19 2014*, ser. CEUR Workshop Proceedings, G. J. Nalepa and J. Baumeister, Eds., vol. 1289, 2014. [Online]. Available: http://ceur-ws.org/Vol-1289/kese10-05_submission_10.pdf

[18] K. Kluza, K. Kutt, and M. Woźniak, "SBVRwiki (tool presentation)," in *Proceedings of 10th Workshop on Knowledge Engineering and Software Engineering (KESE10) co-located with 21st European Conference on Artificial Intelligence (ECAI 2014), Prague, Czech Republic, August 19 2014*, G. J. Nalepa and J. Baumeister, Eds., 2014. [Online]. Available: http://ceur-ws.org/Vol-1289/

[19] I. Weber, J. Hoffmann, and J. Mendling, "Beyond soundness: on the verification of semantic business process models," *Distributed and Parallel Databases*, vol. 27, no. 3, pp. 271–343, 2010.

[12] See: http://www.infoworld.com/slideshow/119652/bossie-awards-2013-the-best-open-source-applications-226975#slide23