Semantic Reference- and Business Process Modeling enables an Automatic Synthesis

Florian Lautenbacher, Bernhard Bauer

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

Abstract. The optimization of business processes is a necessary prerequisite to reduce transactional costs in and between enterprises. Though the modeling of processes is supported by a variety of graphical notations and tools, changes to sub-processes often require the adaptation of the whole process. Thus, methods are necessary to support automated actualization of process models omitting this time-consuming manual task. As a result, business models need to be extended with information describing the semantics of the processes. Machine-understandable information based on standards of the Semantic Web can be applied to automate this task. Describing each process with semantic information enables an automatic synthesis of processes, calculating the optimal combination of them. This paper shows a first approach how to annotate process models with semantic data for a synthesis, describes synthesis algorithms and evaluates a prototypical implementation.

1 Introduction

Over the past few years, enterprises have been undergoing a thorough transformation in reaction to challenges such as globalization, unstable demand, and mass customization. A key to maintain competitiveness is the ability of an enterprise to describe, standardize, and adapt the way it reacts to certain types of business events, and how it interacts with suppliers, partners, competitors, and customers. In the context of process orientation, enterprises today describe these procedures and interactions in terms of business processes and invest huge efforts to describe and standardize them. Business processes are either notated only on a textual basis or graphically with models. During the last decades several graphical model standards emerged like event-driven process chains (EPC) or the Unified Modeling Language (UML2) [11] which is more established in computer science. In particular, the UML 2.0 standard with its extended activity diagrams supports an elegant modeling of business processes.

Nowadays, the creation of business process models can be done with several tools. All of them provide the creation and modification of models, some of them support the users with wizards, but none of these tools currently provides an assistance how the modeled processes should be combined for the optimal flow of the process.

This is, in particular, important for reference processes. Reference process models are the basis for many companies to develop their own business processes. Currently lots of reference processes are available (for a detailed list see e.g. [21]), but each one uses a different language (EPC, UML, OMT, IDEF0, etc.). Additionally, it is very difficult to find a reference process which is applicable for the scope of the business area used in a company. Therefore, these reference processes should be annotated with semantic information to increase their usability and re-use.

An annotation also supports the adaptation of a reference process. Currently, several reference processes are available and companies are using these reference processes and adapting them, but very often after a few years with the changing demands of customers and IT, these reference processes need to be actualized. Annotating the reference processes and the business processes with semantic information, this actualization can be automated. To describe the (reference and business) processes, one can use the annotation that has been developed for several semantic web service standards. These are part of the platform specific model in the model-driven architecture (MDA, [22]) as can be seen in Figure 1. Both, the reference process model (RPM) and the business process model (BPM) would be layered in the Computation Independent Model-layer (CIM). Using similar annotation for RPM and BPM on the one hand and Semantic Web Services in a platform specific view-layer on the other hand will enhance the transformation from CIM to PSM later.

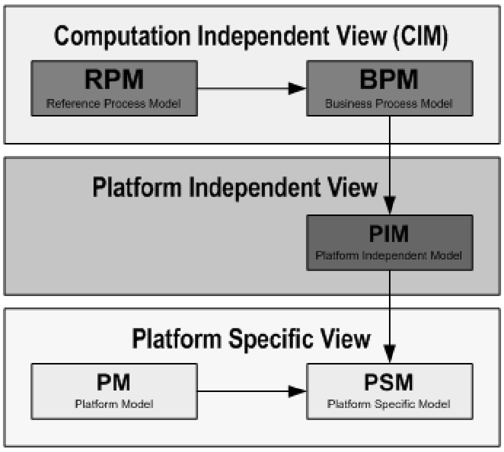


Figure 1: Reference process models in the MDA

The upcoming standards of the semantic web provide a set of concepts that can be used to annotate processes in a way machines can analyze. These concepts are summarized in an ontology which can be built using the resource description framework (RDF), RDF Schema (RDFS), the web ontology language OWL [1] or other standards. Based on these standards web services can be augmented to enable an automatic publication, discovery, interoperability and access. These so-called semantic web service standards (e.g. OWL-S [2] or WSMO [3]) contain parameters for describing the functional behavior of a web service including input and output parameter, preconditions and effects, description of the quality of a service and others. Most of these concepts can be applied to business processes as well.

This paper describes an approach to annotate business and reference processes with machine-understandable information based on standards of the semantic web and semantic web services. Using these metadata a PC is able to compute a (semi-) automatic composition of all processes (called synthesis). Therefore, synthesis algorithms were developed and are surveyed to calculate an optimal composition of all processes (concerning the given parameters).

The paper is organized as follows: Section 2 provides an overview on current approaches of semantic web service standards. The concepts of these standards are the basis to annotate BPM and RPM as described in section 3. In the same section algorithms for the synthesis of business processes are presented and explained. In section 4 a case study based on a prototype implementation is shown. Section 5 gives an overview about related research. Finally, section 6 summarizes the opportunities of our approach and outlines further research activities.

2 Background

In order to obtain a (semi-)automatic synthesis of business processes, processes have to be annotated with semantic information. This requires a language that computers can decode and analyze, i.e. it has to be machine-understandable data. Here the advantages of semantic web standards come into play. Semantic web constructs can be used to describe the data structure as well as the dynamic behavior of an enterprise.

Ontologies provide a basis for describing e.g. context information in a way that human and machines can read and understand. The semantic web stack (cf. [14]) shows the standards that can be used to describe several concepts. Based on RDF and RDFS most ontologies use the standard OWL. But a description of processes needs additional information e.g. what the process does, when it might be started, which inputs it needs, etc. Therefore, several semantic web service standards which describe web services in a similar way can be used to fill this gap: OWL-S, WSMO, SWSF and METEOR-S. In the following these standards will be explained and compared.

OWL-S

The semantic markup for web services (OWL-S, antecessor was DAML-S [15] which is now standardized by W3C) is based on the Web Ontology Language OWL and supplies web service providers with a core set of markup language constructs for describing the properties and capabilities of their web services in an unambiguous, computer-interpretable form. It describes a service in three different ways: ServiceProfile ("What does the service provide?"), ServiceModel ("How is the service used?") and ServiceGrounding ("How does one interact with it?"). In the ServiceProfile the type of the service is specified with the ServiceCategory. A ServiceCategory describes categories of services on the bases of some classification that may be outside OWL-S and possibly outside OWL.

WSMO

Based on the Web Service Modeling Framework (WSMF) [12], the Web Service Modeling Ontology (WSMO) is a formal ontology and language that consists of four different main elements for describing semantic web services:

- Ontologies that provide the terminology used by other elements
- Goals that state the intentions that should be solved by web services
- Web Services which describe various aspects of a service
- Mediators to resolve interoperability problems.

Each of these wsmoTopLevelElements can be described with non-functional properties like creator, creation date, format, language, owner, rights, source, type, etc. WSMO comes along with a modeling language (WSML) [16], a reference implementation (WSMX) and a toolkit (WSMT).

SWSF

Another W3C submission beside OWL-S and WSMO is the Semantic Web Services Framework (SWSF) [4] which represents an attempt to extend the work of OWL-S and consists of two major parts: the Semantic Web Service Language *SWSL* and the ontology *SWSO* above. Both were developed on basis of two different logics: the first-order logic within *FLOWS* (First-order Logic Ontology for Web Services) and based on logic-programming the Rules Ontology for Web Services (*ROWS*).

METEOR-S

Another proposal for a semantic web service standard is the METEOR-S project [5] of the LSDIS-lab, University of Georgia, collaborating with IBM. METEOR-S is the abbreviation of "Managing End-To-End OpeRations for Semantic Web Services" and focuses the process itself. The main point of METEOR-S is the use of semantics in the complete lifecycle of semantic web processes to represent complex interactions between semantic web services.

Comparison

OWL-S provides a distinction between atomic processes, simple and composite processes. This separation is necessary to enable more complex service interactions. The composition of services is abutted on BPEL [25] which is a commonly used language in the web service community and can therefore be easily adapted. The differentiation between ServiceProfile, ServiceModel and ServiceGrounding makes it easier to describe a service completely. The categorization of a service can be used by a semantic-enhanced service registry to find a corresponding service.

The mediator concept of WSMO is very important to solve interoperability issues. Defining goals will be necessary when an automatic orchestration of services is needed. WSMO separates the provider and requester point of view by defining goals and web service capabilities separately (different to OWL-S where no goals can be modeled). It also offers the possibility to add non-functional properties (which lacks in OWL-S) and is not only restricted to web service composition (modeled by transition rules), but also envisions modeling of orchestration.

SWSF extends the OWL-S standard, enables rule languages and extends the description logic used in OWL-S to a first-order logic in FLOWS which makes it easier to describe concepts and their relationships. It already offers the usage of rule languages (similar to WSMO) which is still missing in OWL-S (an extension of the underlying OWL to SWRL in OWL-S could be used to solve this).

METEOR-S separates different ontologies: data semantics for the definition of the vocabulary, functional semantics to describe the capabilities of web services, execution semantics to represent the flow of services and QoS semantics which explains the quality a service offers. This separation makes it easier to create and modify these ontologies.

We will come back to the advantages and concepts of the different standards presenting our own approach.

3 Overall Approach

This section defines our overall approach, how reference and business process models using UML2.0 syntax can be annotated using the concepts of semantic web services. In particular, we explain how an automated synthesis of processes can be achieved using matrices and algorithms working on the annotation.

3.1 Semantic Modeling of Reference and Business Processes

The modeling of business processes concentrates on the functional view whereas web services are focused on the technical view, i.e. different levels of abstraction. Nevertheless, both services and processes can be described with functional (and non-functional) parameters. The semantic web service standards provide descriptions what a service does and how it interacts with others. These descriptions can be applied to business processes and used to describe their choreography.

Each semantic web service standard has advantages when being used to annotate business processes. We are interested in a general approach for the automated synthesis of reference processes, i.e. pre-defined business processes to be customized and combined to obtain value-added business processes. In order to get a long-time realization of the system we decided to combine the concepts and advantages of the standards presented in section 2 in our approach, hence staying independent in the realization being able to switch after it is foreseeable which one is accepted most.

OWL-S provides the distinction between atomic processes, simple and composite processes. Business processes as modeled e.g. in an UML2 activity diagram have different levels of detail. A general manager is only interested in the high level process description whereas a project manager needs a more detailed view. Composing several atomic processes to a composite one enables this requirement. Thus we will keep the distinguished views provided by OWL-S. OWL-S also has a possibility to categorize a service in a taxonomy. This will be important for reference processes, too. **WSMO** enables the definition of goals and several tools are already using the web service descriptions and goal description to create a plan for achieving this goal (e.g. IRS-III [19]). These tools and algorithms might be useful for an improvement of our work, but are currently limited to semantic web services in WSMO. It is the only standard that offers the modeling of non-functional properties which is important for reference and business processes as well. Similar to OWL-S **SWSF** describes each action with four different parameters: input, output, preconditions and effects (in short: IOPEs). This description and abbreviation will be used to describe business processes which will be modeled through actions in UML2.0 activity diagrams. **METEOR-S** separates different ontologies, namely data semantics, functional semantics, execution semantics and QoS semantics. Our approach will use the *data semantics* for the description of the underlying ontology, the *functional semantics* for the description of each process and the *execution semantics* to specify how the processes are related to each other[1].

We adopt and merge the advantages of all standards and build the following architecture which shows how the semantics are interrelated (see Figure 2).

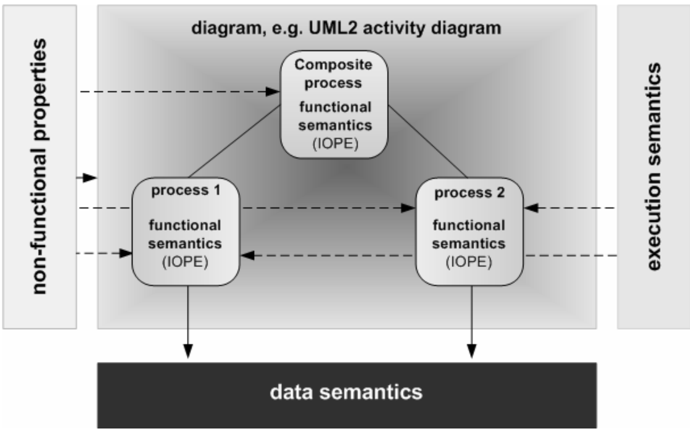


Figure 2: Useful semantics for the synthesis

Our business process model is defined as follows: BPM = (D, F, E, N), whereas D is the data semantics, F are all processes (containing the functional semantics) and relations, E describes the execution semantics and N the non-functional properties. A formal description of a diagram would be $F = \cup_{0 \leq i \leq n} P_i \cup \cup_{0 \leq i, j \leq n} Con_{Pi, Pj}$ where n is the number of processes, P_i is a process and $Con_{Pi,Pj}$ is a connection between the two processes P_i and P_j. A connection $Con_{Pi,Pj}$ describes that after process P_i has been executed, process P_j follows.

The ontology used to define all concepts of the company and corresponding business processes will be called **data semantics** and provides a basis for the modeling of processes. It defines all concepts and their relationships that describe the enterprise, the departments and their tasks which are required to annotate BPM and RPM. Addi-

[1] These data, functional and execution semantics can be modeled using each of the presented semantic web service languages.

tionally, (global) variables can be defined to be used in preconditions and effects for describing the change to a global state, e.g. changes in a database, etc.

Each process is described in the **functional semantics** using IOPEs. Therefore, the concepts defined in the data semantics are used for the input and output section. The variables can be tested for specific values in the preconditions and new values can be assigned in the effects. Each process P_i includes the functional (and non-functional) semantics for this process, $P_i = (I_i, O_i, P_i, E_i, N_i)$. I_i and O_i describe the inputs and outputs of this process and contain concepts that are defined in the data semantics ($I_i \subseteq D$, $O_i \subseteq D$). P_i and E_i are well-formed formulas (of an arbitrary logic) describing preconditions and effects and use variables defined in the data semantics for Boolean expressions and assignments. N_i is the summary of all non-functional properties of this sub-process.

To improve the automatic synthesis it is possible to define the necessary chronology of some processes. Following the METEOR-S approach this is captured in the **execution semantics** E. Using the following statements the user can define his/her preferences for the composition of the processes. Possible statements for two processes P_1 and P_2 are:

- P_1 next P_2 ($P_1 \circ P_2$, means P_2 follows directly on P_1),
- P_1 eventually P_2 ($P_1 \lozenge P_2$, sometime after P_1, P_2 will be executed),
- P_1 previous P_2 ($P_2 \circ P_1$),
- P_1 before P_2 ($P_2 \lozenge P_1$),
- P_1 parallel P_2 ($P_1 \parallel P_2$, means P_1 and P_2 are in two parallel threads).

The **non-functional properties** N can be modeled using a categorization of the service similar to OWL-S and the values in WSMO. The whole process should at least be annotated with the version of the model, when it was created, the name and the subject it is about, the type and format it uses (e.g. UML2) and which business areas it covers. Every sub-process should be annotated with the topic, a short description of the steps in this sub-process, how much it costs and how long it will normally take or after which period of time it must have been finished.

3.2 Synthesis of Semantic Process Models

Having defined all pre-defined processes, i.e. modeled the data semantics, functional semantics and execution semantics, a synthesis task can be started. The non-functional properties are currently not used for the synthesis. These will be used in a future version to optimize the synthesis from an economic point of view.

The synthesis works incremental: first, the functional semantics of each process is compared with the functional semantics of all other processes in a reasoner. The results of these queries are converted to numbers and stored in a synthesis and identity matrix. These matrices are then interpreted as a directed and weighted graph. Based on these graphs the synthesis algorithms can then compute bottom-up what the optimal composition of all processes would look like. We have developed two different synthesis algorithms (Modified Prim and RandomWalk) which will be described in section 3.2.2. The following section describes how the synthesis matrix and identity matrix are built. A more detailed description about the synthesis can be found in [23].

3.2.1 Synthesis Matrix and Identity Matrix

The synthesis and identity matrix are responsible to store the values that are computed in the inference engine. The inference engine has loaded the data semantics and gets queries whether the parameters of two processes fit together. It tests the congruence of the outputs of the first process with the inputs of the second whether they are equal or the inputs are a subset of the outputs. Furthermore they are checking whether the preconditions of the second process are satisfied with the effects of the first. The more these tests are successful, the higher is the value in the synthesis matrix.

The **synthesis matrix *synmat*** is a `n×n` matrix (whereas `n` is the number of processes in the business or reference model) with

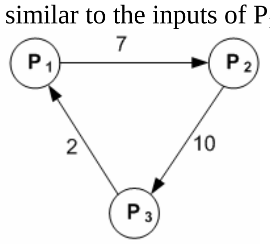
$$synmat_{Pi,Pj} = \begin{cases} 0, \text{ if } (P_i \equiv P_j) \text{ or noMatch}(P_i,P_j), \\ 2, \text{ if Input}(P_j) \subset \text{Output}(P_i), \\ 4, \text{ if Input}(P_j) = \text{Output}(P_i), \\ 5, \text{ if Preconditions}(P_j) \subset \text{Effects}(P_i), \\ 6, \text{ if Preconditions}(P_j) = \text{Effects}(P_i), \\ 7, \text{ if } (\text{Input}(P_j) \subset \text{Output}(P_i)) \wedge (\text{Preconditions}(P_j) \subset \text{Effects}(P_i)), \\ 8, \text{ if } (\text{Input}(P_j) \subset \text{Output}(P_i)) \wedge (\text{Preconditions}(P_j) = \text{Effects}(P_i)), \\ 9, \text{ if } (\text{Input}(P_j) = \text{Output}(P_i)) \wedge (\text{Preconditions}(P_j) \subset \text{Effects}(P_i)) \\ 10, \text{ if } (\text{Input}(P_j) = \text{Output}(P_i)) \wedge (\text{Preconditions}(P_j) = \text{Effects}(P_i)). \end{cases}$$

We have chosen these numbers for simplification of the implementation: every entry in the matrix bigger than six can be created by adding the smaller numbers and is unambiguously factorable into these smaller numbers.

In the example in Figure 3a the outputs and effects of process P_1 are sufficient (but not equal) for the inputs and preconditions of process P_2 (a value of 7). The outputs and effects of process P_2 are equal to the inputs and preconditions of process P_3 (value: 10). Additionally the outputs of P_3 are similar to the inputs of P_1 (value of 2).

	P_1	P_2	P_3
P_1	0	7	0
P_2	0	0	10
P_3	2	0	0

(a)



(b)

Figure 3: A synthesis matrix and the corresponding graph

Similar to the synthesis matrix the **identity matrix *idenmat*** contains the results of the comparison of two processes, but this time the equivalence of them is tested. If parts or all inputs are the same (alternatively outputs, preconditions or effects) there is a value bigger 0 (maximum 16), otherwise with a value of 0 the two processes are not parallelizable or the process was compared with itself.

3.2.2 Modified Prim and RandomWalk

Having created both matrices these can be interpreted as a directed and weighted graph G = (N, E). The nodes N are the processes and there are edges if the entries in the synthesis matrix between two processes are bigger than 0 (cf. Figure 3b). Based on these graphs the synthesis algorithm can be started.

Modified Prim is an adapted Prim- (or Dijkstra-) algorithm and creates a graph choosing one edge randomly first. Then the edge with the highest value will be added to the solution if it fulfills given constraints. This is repeated until all processes have been visited. After visiting each process, the operation is terminated and the solution is evaluated. This algorithm is executed with each existing edge as start edge and after all solutions are computed, the one with the best rating is returned to the user.

The synthesis algorithm **RandomWalk** operates on basis of an existing solution (e.g. generated via Modified Prim). RandomWalk was primarily invented to optimize mathematical problems and tries to converge to a local (or better: global) maximum by changing the solution iterative. Every solution is rated (using the value of the applied edges) and computed whether there exist solutions with a better rating. Therefore, in our case, edges are removed by accident and others (with a better value) are added. After a period of time the operation is stopped and the so far best solution is returned which represents the best computed workflow of the annotated processes. Of course, it is also possible to specify a goal (this would be the effects or outputs of the whole activity) and the algorithm will reject all solutions which don't give the required outputs or effects.

4 Case Study

To test the semantic process modeling and to compare the different synthesis operations, a prototypical tool was developed [6]. It offers the modeling of a semantically-enriched UML2 activity diagram and testing the synthesis with the operations explained above.[2] The big advantage of the synthesis comes to the fore using it on extensive reference process models. However, due to space limitations we only demonstrate the synthesis of a small business process.

Let us assume a purchase process where a customer buys a product which has to be adapted to his needs. First, the warehouse has to be checked whether the necessary raw product is available or perhaps a comparable product can be found. If this is not the case, then the raw product is ordered. With the raw product the end product can be manufactured, a bill can be written and the product can be delivered to the customer. Without semantic enhancements, the system would not be able to decide whether the process "Order product" or "Write bill" should come first. Therefore every process can be annotated with inputs, outputs, preconditions and effects.

A part of the semantics for this example would be the following: Figure 4 shows a part of the data semantics for one of these processes ("Order product"). The functional semantics for the same process would include (informal):

Precondition: ProductInStock hasValue FALSE,
Output: Order,
Effect: ProductOrdered setValue TRUE.

[2] The case study is based on OWL and OWL-S to present the ontologies, but every concept can be adapted to other semantic web service languages, too.

```
<semBPM:Object rdf:ID="Order"/>
<semBPM:SemDataProperty rdf:ID="executedFor">
        <rdfs:domain rdf:resource="#Order"/>
        <rdfs:range rdf:resource="#Customer"/>
</semBPM:SemDataProperty>
<semBPM:globalVariable rdf:ID="ProductInStock"/>
```

Figure 4: Data semantics for the process “Order product”

Having modeled all processes (without transitions!), annotated them with semantics and started the synthesis, one gets the result of Figure 5. Both algorithms achieve the same result and the solution the user might have probably expected. If one of these processes changes or needs to be deleted, one can simply start a new synthesis to get the new optimal combination and no further action is required.

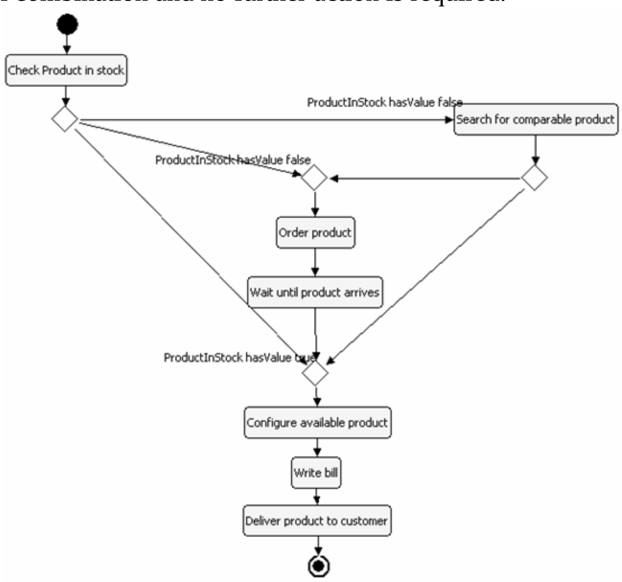


Figure 5: Result of the synthesis of our business trip example

Testing more complex cases the algorithms sometimes find different solutions which are syntactically and semantically correct, but not always the optimal and expected results concerning the given parameters. The creation of the synthesis and identity matrix has a time and space complexity of $O(n^2)$. We are currently working on optimizing Modified Prim, which has a time complexity of $O(n^3)$, whereas RandomWalk only needs $O(n \cdot \log n)$.

5 Related Work

This is – to our knowledge – the first approach to annotate business processes and use these annotations to make an automatic synthesis of the modeled processes. Other institutions are describing business processes as well: [24] describes an approach to use ontologies for business process modeling in Petri nets. Jenz & Partner is developing a tool named BPEdit [9] which enables users to graphically define and edit business process definitions using the Business Process Modeling Notation (BPMN) [8]. BPEdit relies on an open and extensible ontology-based information model which is defined in OWL and offers the generation of deployable process definitions in process definition languages such as BPEL. The METEOR-S development team is developing a Process Designer [10] to generate BPEL Web Processes based on the METEOR-S Web Service Annotation Framework (MWSAF) [13] and which supports dynamic discovery of partner services using semantics. Both tools enable the annotation of business processes and generate process definitions in BPEL. They use semantic web languages like OWL and develop concepts to annotate business processes with semantic information. But at present they are not using these annotations to make an automatic synthesis of the described processes.

There are several groups working on composition of web services: they either require static information (e.g. Golog-based [17] or based on Hierarchical Task Networks [18]). Some approaches use ontologies [19], others agent technologies [20]. But none of them currently considers business processes and the automatic composition of UML2 activity diagrams.

6 Conclusions and Outlook

The presented approach is a first step to use annotated business process models to make an automatic synthesis and compute the best composition of the modeled processes. This enables a modeling tool to assist humans in modeling and optimizing a model when changes to the corresponding actions have occurred. Our approach enables the usage of ontologies to describe business processes and to use this information to compute an optimal workflow of business processes which is necessary for an automatic adaptation of reference processes and will increase their utilisability and usage.

Our approach combines the advantages of the discussed semantic web service standards, but stays adaptable to each of these standards. We implemented a prototypical application which was developed as an eclipse plug-in and provides therefore an opportunity to be connected with other plug-ins.

Our prototype only offers the modeling of UML2 activity diagrams; other diagrams are currently not supported. Especially diagrams to cover organizational or resource aspects should be included and additional semantics and ontologies created (e.g. for the organizational structure of the company). We will create a UML2 profile for the usage of ontologies in business process models and try to combine the higher-level business process descriptions with lower-level web services. The synthesis will be extended not to consider optional elements and to use actions (e.g. for consistency checking) several times.

We are aiming to develop a methodology to annotate reference processes, whereas we will consider current approaches to model business processes like BPMN and approaches of business process ontologies.

References

1. Guiness, D. and Harmelen, F.: "OWL Web Ontology Language - Overview", February 2004, W3C Recommendation.
2. Martin, D. et al: "OWL-S: Semantic Markup for Web Services", November 2004, W3C Member Submission.
3. Lausen, H., Polleres, A. and Roman, D. (Eds.): "Web Service Modeling Ontology (WSMO)", June 2005, W3C Member Submission
4. Battle, S. et al.: "Semantic Web Services Framework (SWSF) Overview", September 2005, W3C Member Submission.
5. LSDIS, University of Georgia, "METEOR-S: Semantic Web Services and Processes", Homepage: http://lsdis.cs.uga.edu/projects/meteor-s/
6. University of Augsburg, Plug-In available at http://pvs.informatik.uni-augsburg.de/eclipse
7. Axenath, B., Kindler, E. and Rubin, V. "the Aspects of Business Processes: An Open and Formalism Independent Ontology", Paderborn, April 2005.
8. BPMI.org, "Business Process Modeling Notation (BPMN) – Version 1.0", May 2004.
9. Jenz & Partner GmbH: "BPEdit – What is it?" available online at http://wa0529.dw10.de/MamboV4.5.2/content/view/15/28/
10. METEOR-S Download and Release Page: "METEOR-S Process Designer" available online at http://lsdis.cs.uga.edu/projects/meteor-s/downloads/index.php?page=4
11. Object Management Group (OMG): "Unified Modeling Language (UML) Specification".
12. Fensel, D., Bussler, C.: "Web Service Modeling Framework WSMF", ECommerce, 2002.
13. LSDIS, University of Georgia, "MWSAF: METEOR-S Web Service Annotation Framework", Homepage: http://lsdis.cs.uga.edu/projects/meteor-s/mwsaf/
14. Berners-Lee, T. "Putting the Web back in Semantic Web", In: ISWC, Galway, 2005.
15. The DAML Services Coalition: "DAML-S: Semantic Markup for Web Services", 2003.
16. deBrujin, J., Lausen, H.: "Web Service Modeling Language", W3C Submission, 2005.
17. McIlraith, S. and Son, T.: "Adapting Golog for composition of semantic web services". In: KR-2002, Toulouse, France, 2002
18. Au, T.C., Kuter, U. and Nau, D.: "Web Service Composition with Volatile Information", In: ISWC05, Galway, November 2005.
19. Domingue, J., Galizia, S. and Carbal, L.: "Choreography in IRS-III – Coping with Hetereogeneous Interaction Patterns in Web Services", In: ISWC05, Galway, 2005.
20. Greenwood, D., Buhler, P. and Reitbauer, A.: "Web Service Discovery and Composition using the Web Service Integration Gateway", in: EEE'05.
21. Fettke, P. et al.: "Business Process Reference Models: Survey and Classification", BPRM05, Nancy, September 2005
22. Miller J. and Mukerji J. (Eds.): "MDA Guide Version 1.0.1", June 2003, omg/2003-06-01.
23. Lautenbacher, F., Bauer B. "Automatic Synthesis of Reference Processes Using Semantic Concepts", Poster at I-ESA06, Bordeaux, March 2006, available online at www.ds-lab.org/publications/proceedings/2006_I-ESA2.html
24. Koschmider, A. and Oberweis, A.: "Ontology based Business Process Description", in: EMOI-INTEROP'05, Porto, June 2005.
25. OASIS: Business Process Execution Language For Web Services, Version 1.1, 5 May 2003.