Semantic Process Modeling – Design and Implementation of an Ontology-based Representation of Business Processes

In the article, an extension of process modeling languages is designed and realized as a prototype. With it, the semantics of labels for process model elements formulated in natural language can be represented by terms from a formal ontology. This formalization of model element-related semantics can eliminate the scope for interpretation connected with the use of natural language and improve query functionalities in modeling tools. In addition, new possibilities for the validation of process models are provided through the use of rules in combination with ontologies.

The Authors

Prof. Dr. Oliver Thomas
Michael Fellmann M.A.
University of Osnabrück
Institute of Information Management
and Corporate Governance, Chair
in Information Management
and Information Systems
Katharinenstraße 3
49074 Osnabrück
Germany
{oliver.thomas | michael.
fellmann}@uni-osnabrueck.de

1 Introduction

A variety of modeling languages are used for planning and controlling business processes in research and practice. Examples are the Petri net, the event-driven process chain (EPC), the UML-activity diagram or the Business Process Modeling Notation (BPMN). In modeling languages semi-formal, graphic forms of representations have gained acceptance, which are closely related to technical terms from the field of business, but exact enough so that models can serve as a starting point for the implementation of application systems. Although this is a fundamental idea in model-driven development, one of the main problems for modelers lies in the above connection of natural language with the graphic form of representation: the labels of individual elements in a business process model, for example "Check order" as an label for an EPC function or a BPMN activity are added by the modeler in a natural language, regardless of the decision made for a description language. Thus, a fundamental part of the semantics of a process model is always tied to natural language. This results in two significant classes of problems in the development and application of models.

First, naming of a model element by use of natural language brings scope for interpretation along. This is referred to in literature as a linguistic or term defect; among these are for example, synonyms, homonyms, equipollences, vagueness and false signifiers (Ortner 1997, pp. 31 ff). These term defects lead to problems that limit the use of models as a medium for communication and as starting points for IT-concepts. If, for example, a model element in a model is referred to as "merchandise" and a further element in a different model as a "good" and both artifacts represent the same object then a synonymy has occurred and with it the risk of not recognizing equivalent constructs that could lead to a duplication of efforts when models are translated from the level of requirements definition to design specifications (Rosemann 1996, p. 188).

And *second*, in addition to linguistic defects, the use of natural language implies semantics of process models which are not machine-processable. This leads to difficulties when querying database-backed modeling tools, as well as validating process models. The problem with queries concerning modeling tools is that the retrieval of facts about processes which have not been explicitly recorded by the model's constructor, but are nevertheless derivable with the help of logical conclusions, is not possible. An example for this is a function that accesses resources in stock. If these connections are not specified in a machine-processable manner, then it cannot be concluded, for instance, that the above function reduces the stock. A further problem is that the validation, understood as the question of whether a model serves the purpose it was created for, is not possible without machine processable semantics (Desel 2002, p. 24). As a result this means that inadequate models cannot be recognized while being developed.

In this article, both of the above problems, which result from the use of natural language when naming process model elements, are solved by an extension of semi-formal modeling languages using ontologies. The basic idea for this extension, which we will call *semantic process modeling*, is the assignment of concepts for-

BANG &
OLUFSEN
RUNS
SAP.

THE BEST-RUN BUSINESSES RUN SAP™

SAP®

malized in ontologies to process model elements. Thus, the semantics of process model element labels formulated in natural language can be represented formally and in a machine-processible way.

To achieve this goal we will proceed in the following manner: first, related work will be discussed and semantic problems in business process management studied, whereby contributions from related disciplines shall also be consulted (section 2). To close the research gap we will then develop a method which will be used for the derivation of the semantic process modeling (section 3). This derivation takes place in two steps. In the first step, the approach will be illustrated using selected modeling and ontology languages as well as generalized with the help of an information model (section 4). In the second step, the general usability of the developed approach will be demonstrated based on an implemented test environment (section 5). The article closes with a critical discussion of the applicability of the approach (section 6) and an outlook on future research tasks (section 7).

2 State of research and related work

While the term semantics often refers to the meaning or content of a word or sentence in everyday language, the scientific meaning refers to the conception of linguistics. In this discipline, semantics refers to the branch that deals with the meaning and significance of language resp. linguistic signs. In other words: the teaching of the meaning and the relations of signs for a certain object. If this is transferred to process modeling languages, the semantics of a process model – in a first approach – can be understood as the relationship between the elements of a model (sign) and an existing or future operational business process (universe of discourse). However, this understanding of semantics is not shared by all authors. In early works on process modeling (e. g. Keller et al. 1992), the adjective "semantic" was used to emphasize the importance of the modeling of business processes from a business, non-technical perspective.

Up to now, studies on the semantics of modeling languages have concentrated primarily on the *formal* semantics of available language elements – in the following also referred to as language constructs.

Formal semantics is anchored in several disciplines, such as theoretical informatics or logic and deals with the exact meaning of artificial (i. e. constructed) or natural languages. Mathematical methods are important when studying the formal semantics of modeling languages (for the EPC cp. Kindler 2006 and the literature quoted there). However, the semantics which is added to model elements in the form of model element labels and which is, especially in the case of semi-formal languages, based on natural language, is not accounted for in these works.

Today, the formal representation of knowledge in the field of Artificial Intelligence is being advanced irrespective of the development of modeling languages as presently used for designing information systems. Ontologies are often used to represent these complex knowledge relationships and are gaining in importance due to efforts in extending the World Wide Web to a Semantic Web (Fensel et al. 2003). In the informatics-related interpretation according to Gruber (1993, p. 199) used here, an ontology is understood as an explicit formal specification of a conceptualization, which is an abstract, simplified view of the world that can be represented to serve certain purposes (Gruber 1993, p. 199). Newer attempts to find a definition also emphasize the inter-subjective validity of the conceptualization, so that ontologies can be seen as a vocabulary developed by several individuals and accepted and used by a group (shared conceptualization) (Studer et al. 1998, p. 186; Gómez-Pérez et al. 2004, p. 8). Information Systems uses the comprehensive preliminary work in ontology research, for example in the field of knowledge management (Zelewski et al. 2005), product data management (Hahn 2005) or information retrieval (Kuropka 2004). In these studies structural and hierarchically organized artifacts (for example, documents or product models) are represented by ontologies. The use of these representations for the semantic enrichment of business process models is the topic of this paper.

With regard to ontological analyses and in the international context of the information systems discipline, the Bunge-Wand-Weber-model should, in particular, be emphasized (Wand and Weber 1995). Put simply, it can be understood as a theory for the description of information systems. Although the model is not suited to record all of the phenomena in the design

of information systems, it is often used in research for the ontological evaluation of modeling tools (Green and Rosemann 2000) and methods (Green 1996), of interoperability standards (Green et al. 2005), as well as for the selection and introduction of standard software (Soffer et al. 2001). The evaluation of the suitability and requirements of information models connected with this and the modeling languages used for their construction as well as the derivation of the criteria for determining the quality of the artifacts are not the subject of the approach suggested in this paper. In contrast to the use of ontologies for evaluating modeling language artifacts, the semantic process modeling is based on the representation of the artifacts in an ontology.

The potential of such a connection between ontologies and process models has been recognized in literature for quite a while now (Hepp et al. 2005; Lin and Strasunskas 2005; Ahlemann et al. 2006; Hepp and Roman 2007). Usually, the authors try to create the foundation for the automated processing of procedure models with the ontology-based attribution of process models. Most of these works are geared toward a certain language, i. e. they only deal with the semantic annotation of process models represented with the help of a certain language. Such semantic extensions for process description languages exist for example for the Petri net (Koschmider and Ried 2005; Brockmans et al. 2006), the EPC (Thomas and Fellmann 2007; Bögl et al. 2008), the BPMN (Abramowicz et al. 2007), the Demo Engineering Methodology for Organizations (DEMO) (Dietz 2006) and the Extended Enterprise Modeling Language (EEML) (Lin and Ding 2005). There is also an approach for the automatic synthesis and modification of models after changes to sub-processes for the UML-activity diagram (Lautenbacher and Bauer 2006). Sometimes the semantic annotation of process models leads to the definition of new languages, such as in the case of the Process Semantic Annotation Model (PSAM) from Lin (2008). This last author also developed the most comprehensive study on the semantic annotation of process models with the Process Semantic Annotation Tool. The approach to the annotation of business process models described in this paper is language-independent in contrast to the language-specific approaches. In com-

parison with the paper mentioned above that allows for the multiple annotation of a process model element for the description of the tasks or goals represented by the element, the annotation of a model element with only one ontology element will be proposed in this paper. We will go into detail about this later.

The standardization of terminology used in models by means of a semantic annotation can also be achieved by using technical terminology models (Ortner 1997; Rosemann and Schwegmann 2002). The use of ontologies for the formal representation of a domain has, however, the advantage of being machine-interpretable. This means that facts not represented explicitly can be automatically complemented by methods of machine inference, in order to interpret the semantics completely when, for example, searching for or validating models.

This machine-processability is also the fundamental idea of research in the field of Semantic Web Services (Cabral et al. 2004; Cardoso and Sheth 2005). The semantic description of web services aims at the improved selection and orchestration of these services and is thus tailored to the specifics of service-oriented software development. In contrast to these works, our approach deals exclusively with business process models described on an expert level. This accommodates the fact that organizational difficulties still pose one of the most frequent obstacles in the development and modification of business processes (AIIM 2007).

A corresponding approach to the automated planning of expert process models was developed in the project SEMPRO (Henneberger et al. 2008; Heinrich et al. 2008). Here, suggestions are automatically made for complete process models on the basis of individual process actions or fragments as well as a start and target state. The semantic description of process actions, via their input and output parameters, required for this procedure can be realized with the approach developed here for connecting ontologies and process models. This is, however, not tied to a certain application, but rather constitutes a foundation for developing approaches on process planning, improving queries to process databases or extending the validation of process models.

Fig. 1 Extended semiotic triangle for the semantic process modeling
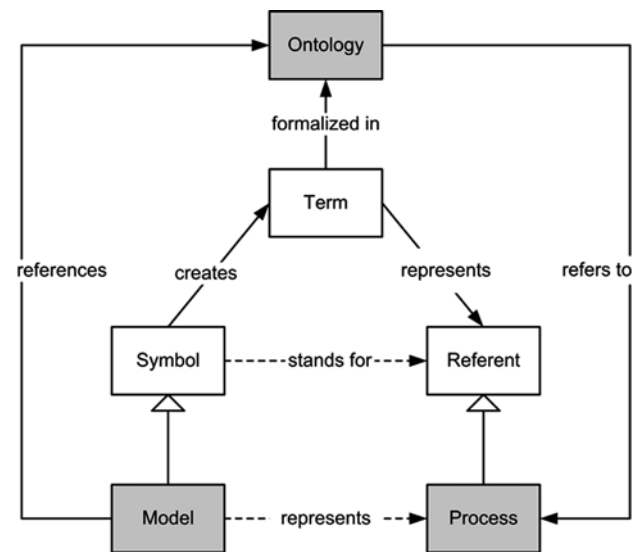
3 Method and procedure

The central thought in semantic process modeling is based on an extension of the semiotic triangle. In its basic form according to Ogden and Richards (1923) it states that symbols have no direct and immediate relation to the things for which they stand (inner triangle, **Fig. 1**). In fact, a relation is only indirectly possible through terms that refer to the things and which are activated by the appearance of symbols. This means that the interpretation or perception of things through a subject is only possible when it has a term, concept or a mental model associated with the symbol.

Analogous relations arise between the elements of the extended semiotic triangle "model", "ontology" and "process" (outer triangle, **Fig. 1**). Models are interpretable as a special type of symbol. A relationship between a model and a process is not directly and immediately possible, because the model is the result of a construction process characterized by creativity. The meaning of the model elements depends on the idea resp. the mental model that the model's developer has. This is especially true for the label given to model elements used as symbols for the terms applied during the construction phase. Through the connection of process model elements with terms from formal ontologies, indicated by the arrow "references" in **Fig. 1**, the semantics formulated with natural language and contained in the model element label can be explicitly specified.

If for example a model element with the symbol "Customer order" is found, then the explicit conceptualization in the ontology formalized by domain experts can be consulted for its interpretation. This provides more information about the identified term and relates it to other terms. Thus, one can define in the ontology that a customer order has a date and is a more specific term than "Order", which in turn is a more specific term than "Document". These connections can be processed by a machine, and this shows a central aspect of the semantic extension of process models with ontologies: the knowledge represented in ontologies can be used for the machine interpretation of process models. Through the method of machine inference new facts not explicitly contained in the original model are generated. The query for finding a process model with the element "Process order" can thus detect a process model with the element "Process customer order" as a result, because the information stating that customer order processing is a specialization of order processing is contained in an ontology in machine-interpretable form.

In order to put the extension of process modeling demonstrated with the extended semiotic triangle into practice, languages for representing ontologies (ontology languages) and process models (modeling languages) must be selected. Furthermore an ontology has to be (re)constructed. In a next step, a special annotation of the process models on the basis of the selected languages must be derived as well as generalized in a cross-language manner.
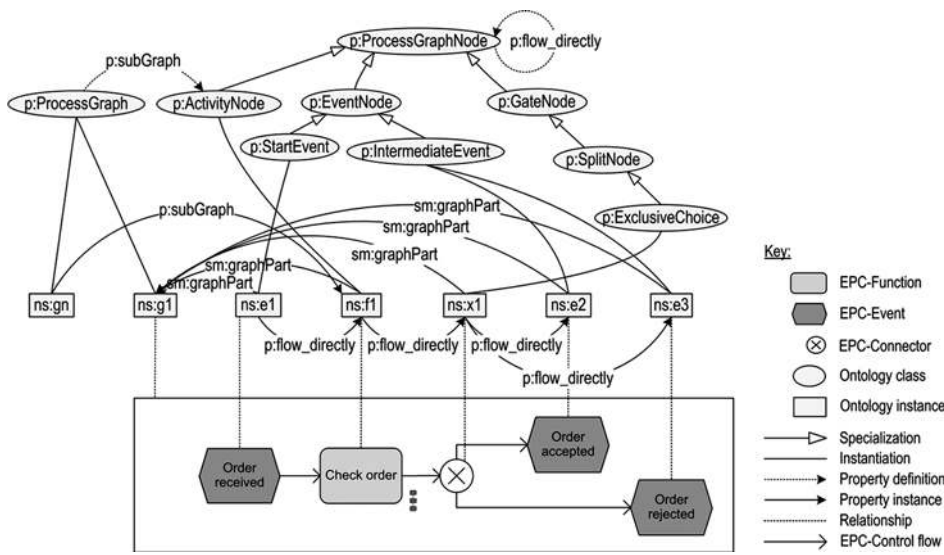
Fig. 2 Ontology-based representation of a process model based on EPC

4 Conception for semantic process modeling

4.1 Ontologies and ontology languages for process management

Semantic process modeling assumes that ontologies and ontology languages exist that can be used for representing the semantics of model elements. To construct an ontology for the semantic process modeling, existing ontologies such as the Enterprise Ontology (Uschold et al. 1998) or TOVE (TOronto Virtual Enterprise) (Fox 1992) can be applied. Further ontologies have become available through the translation of established common standards into ontology languages such as, for instance, eClassOWL as a porting of the eCl@ss-standards to OWL (Hepp 2005) or the generation of ontologies from semantic structures already existing in the company such as for example relation models (Gómez-Pérez and Manzano-Macho 2003). If terms from several ontologies are used then the resulting ontology can be further structured by means of a comprehensive upper ontology (cp. the overview provided by Gómez-Pérez et al. 2004, pp. 204 ff). For the ontology construction in this article we have selected SUMO (Suggested Upper Merged Ontology) (Niles and Pease 2001), because it has a comprehensive, hierarchical category system, which in comparison to abstract upper ontologies allows the easier structural classification of the terms to be defined into the existing ontology.

Many ontology languages are available for the explicit and formal representation of an ontology (cp. the overview provided by Gómez-Pérez et al. 2004, pp. 204 ff). The ontology language OWL (Web Ontology Language) is especially relevant for the semantic process modeling approach in this article due to its comprehensive acceptance and tool support as it is widely used outside the AI-research community and as it is standardized by the W3C (Smith et al. 2004). Thus, it has been selected as our ontology language in this article. Because OWL exists in the three variations Light, DL and Full, a suitable sub-language has to be selected. Due to the emphasis on the machine-processibility of semantics by the semantic process modeling, we selected OWL DL which is based on description logics. It is based on the description logic SHOIN(D) and comprises negation, disjunction and a limited form of existential and universal quantifiers. In contrast to OWL Full, powerful inference machines exist for OWL DL, such as for example Pellet (http://pellet.owldl.com), FACT++ (http://owl.man.ac.uk/) and Racer (http://www.racer-systems.com/). In the following, we will use the term "OWL" instead of the term "OWL DL" for reasons of linguistic simplicity. For criteria-based selection of ontology languages we refer to further readings (Gómez-Pérez et al. 2004, Casely-Hayford 2005).

4.2 Ontology-based process representation

A prerequisite for the ontology-based process representation is the specification of concepts in the ontology which correspond to the language constructs (e. g. EPC-function) which have been used to create a process model. Individual model elements can then be represented by instantiating these concepts (described in this section). This representation makes it possible to add further statements about model elements by connecting them with other terms in the ontology (paragraph 4.3).

Terms and relations are basic elements provided through ontology languages irrespective of underlying formalities. In OWL they are also referred to as classes and properties. The form of the representation is relatively free. Thus for example, one must decide whether relationships between language constructs should be represented in the ontology as properties or as instances of classes. In this article, process models are interpreted graph-theoretically, i. e. a process model is understood as a directed graph with nodes and edges. Types of nodes, such as an EPC-function or a BPMN-activity, are defined as classes in the ontology. Relationships between these types of nodes, for example in the form of an EPC-control or BPMN-sequence flow, are defined as properties in the ontology. This procedure allows the simple and intuitive representation of process models in ontologies whose limitations, however, lie in the fact that the possibilities for defining relationships between language constructs depend heavily on the possibilities of the selected ontology language.

Fig. 2 exemplifies the outlined, ontology-based process representation by depicting a process model fragment created with EPC. The events, function and connector of the EPC-model are represented in the ontology by instances of
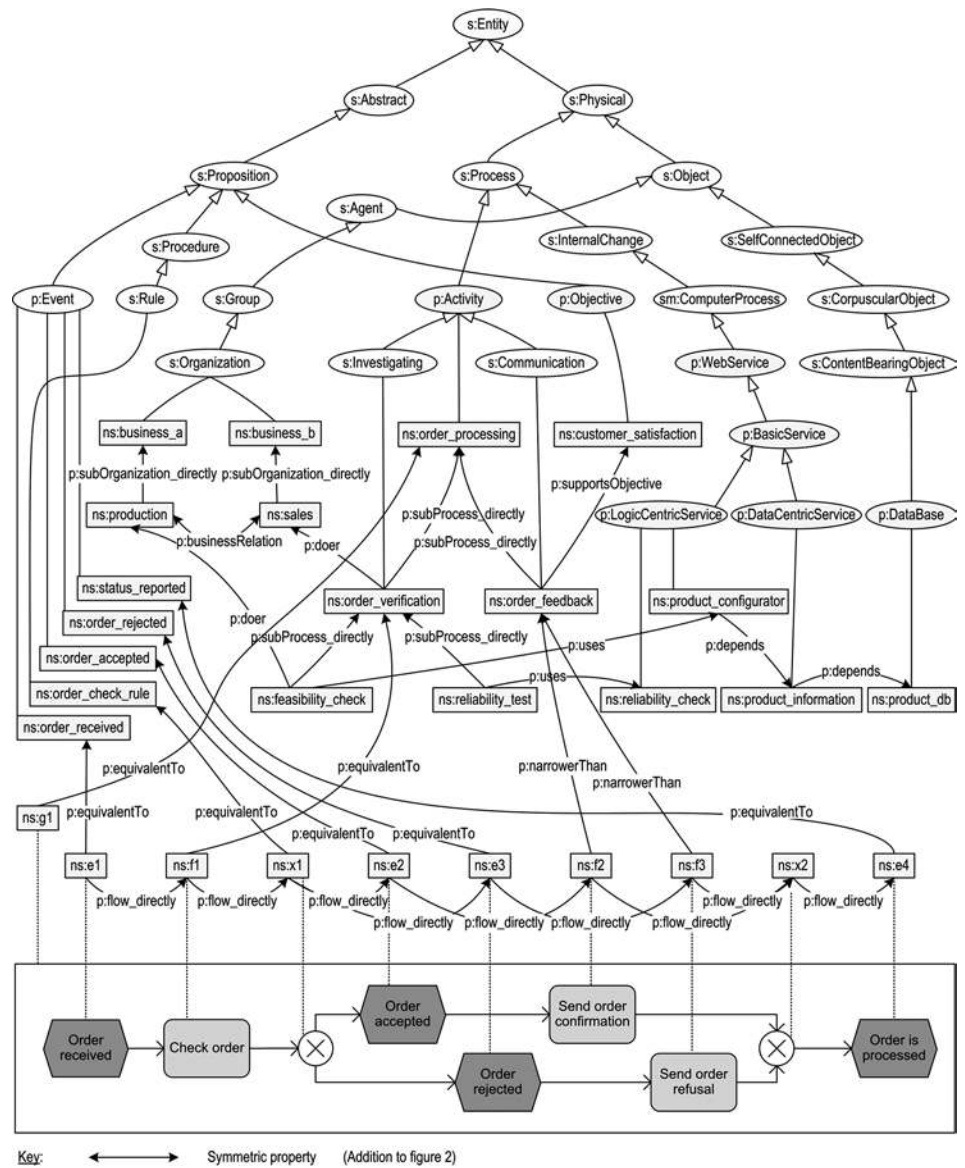
Fig. 3 Semantic annotation of an EPC-model

classes, which – due to the graph-theoretic interpretation – are derived from the class *ProcessGraphNode*. It should be noted that the prefixes placed in front of the ontology element names indicate namespaces (Bray et al. 2006). Thereby the prefix "ns" represents any namespace in which the classes and instances of the exemplified ontology are located. The prefix "s" references the namespace of the SUMO-ontology, "p" the namespace of the SUMO-extension used here, and "sm" the namespace of the SUMO Mid Level Ontology (Niles and Pease 2001). In the interest of readability we will omit the namespace prefixes in the further continuous text. The ontology classes used for the representation of the EPC-connectors are defined according to studies on the language-independent description of workflow patterns (van

der Aalst et al. 2003). The chronological and logical connection existing between these model elements, which is called control flow in the EPC, is represented in the ontology by the property *flow_directly*.

4.3 The annotation of process models

The connection of the process models and model elements with other elements in the same ontology is called *semantic annotation*. The attribute "semantic" emphasizes – in contrast to an annotation for example in the form of a freely formulated text – the claim for the machine-processibility of additional information given by the annotation. The following explanation refers to the annotation of model *elements* which could, however, also be transferred

to the annotation of whole models in the same manner.

An important characteristic of the annotation is the cardinality of the connection of represented model elements (in the following also referred to as "model element instances") with other instances in the ontology (for better understanding referred to as "domain instances" in the following). A 1:n-relationship exists when a model element instance, such as "Check order" is connected via suitable relations with several domain instances such as, for example, "Sales" for the documentation of the executing organizational unit and "Product configurator" for checking the technical feasibility of an order. The multiple connections of model element instances with domain instances leads, however, to redundant annotations when
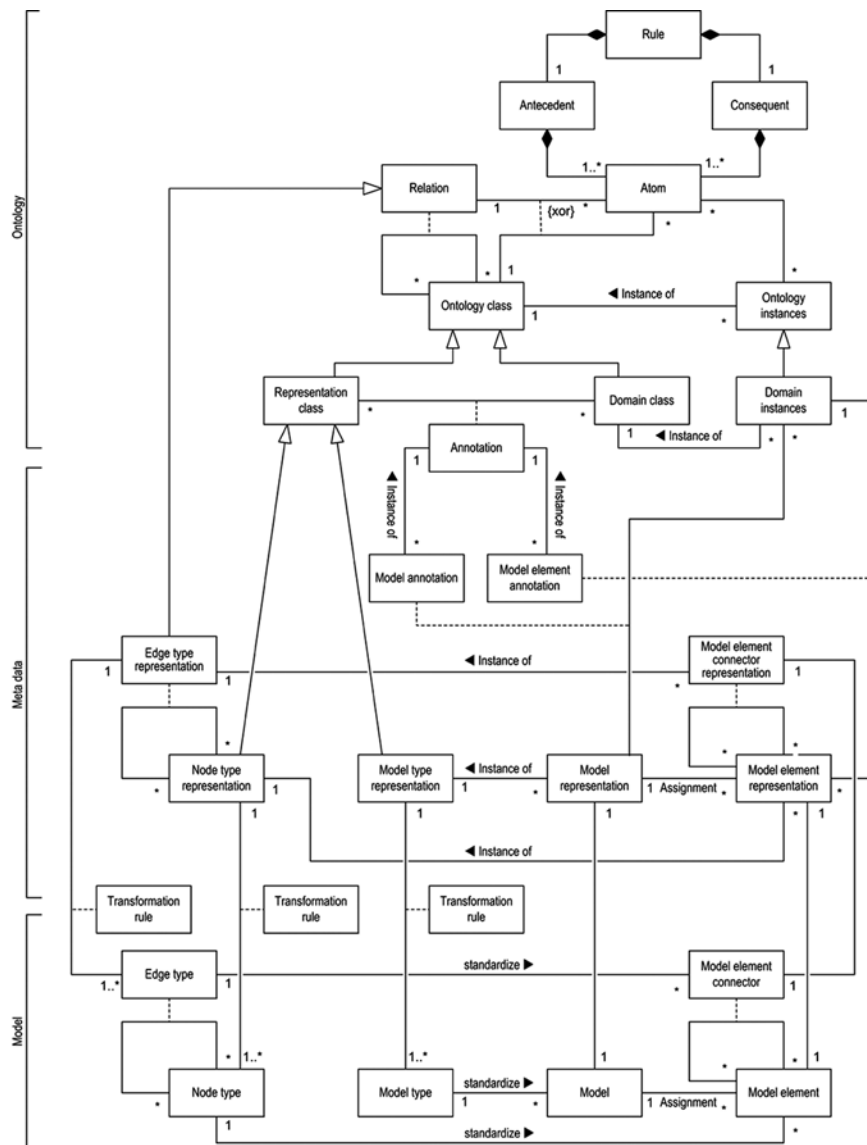
Fig. 4 Information model for the semantic annotation of process models

ment instance semantically, the model element instance, however, is defined more specifically or narrower in certain aspects. An example for this is the domain instance "Collect data" and the model element instance "Collect customer data". Through this relation, several model element instances can be annotated with *one* domain instance, which reduces the number of required domain instances. In addition, an annotation with this relation is even possible when no equivalent domain instance exists for a model element instance at the time of the annotation. In this case, the model element instance can be assigned a more unspecific domain instance through narrowerThan. Thus, in the example in **Fig. 3**, the two model element instances f2 ("Send order confirmation") and f3 ("Send order refusal") are assigned to the domain instance *order_feedback via narrowerThan*.

All in all, **Fig. 3** shows the annotation of an EPC-model using the relations equivalentTo and narrowerThan in an exemplary manner. The ontology classes and their relationships used in the top part of the figure are based on SUMO (cp. section 4.1).

4.4 Integration of rules

A connection of process models with ontologies is possible with the described representation and annotation. The definitions contained in the ontology can be used for terminological standardization, but also for queries and validation. However, all of the relevant connections in a domain cannot be represented by defining terms and relations. An extension to the use of rules is necessary especially for the representation and conclusion of complex dependencies between several terms.

The rules necessary within the framework of the semantic process modeling are divided into deductive and normative rules based on Boley et al. (2007, pp. 273 ff). Deductive rules, also referred to as derivation rules in the field of business rules, help in deriving new facts on the basis of existing facts through the use of logical implications. Thus for example, the following IF-THEN-rule can be formulated in the ontology during the interpretation of the process from **Fig. 3** under the premises that the goals supported by a sub-process also belong to the group of goals supported by the super-process: "*If the two activities x and y exist and x is*

semantically equal model elements occur in several models. Therefore, in this article we suggest a 1:1 relation, i. e. the annotation of a model element instance with only *one* domain instance that represents it semantically. An example for this is the model element instance "Check order" connected with the domain instance "Order check" in the ontology. This procedure of annotation principally requires more domain instances (if the elements of the process models to be annotated are semantically fully disjunctive, then a maximum of one corresponding domain instance can be necessary for each model element to be annotated). The number of domain instances in the ontology can be reduced by splitting the annotation relation into a relation of semantic equivalence and a relation with a higher semantic

specificity following the description of the relations between elements of controlled vocabularies (Miles and Bechhofer 2008).

A relation of semantic equivalence, referred to as equivalentTo, exists between a domain instance and a model element instance when the former exactly reflects the indented semantics of the model element instance (i. e. the semantics of the object which the construction of the semiformal model was originally based upon). An example for this can be seen in **Fig. 3** in the annotation of a model instance f1 ("Check order") with the representing domain instance *order_verification*.

A relation with a higher semantic specificity, referred to as narrowerThan, exists between a domain instance and a model element instance when the domain instance contains the model ele-

Deutsche Telekom Laboratories

We shape the future

Deutsche Telekom Laboratories is Deutsche Telekom's research and development institute based in Berlin. It is simultaneously a scientific institute organized under private law and associated with the Technische Universität (TU) Berlin. At Deutsche Telekom Laboratories, scientists from across the globe work together with experts from the Group to develop new services and solutions for Deutsche Telekom's customers. Establishing new companies (spin-offs) is another method for making use of research output.

Cooperation with the TU Berlin, other universities and industry partners creates a bridge between business and science in order to turn ideas into marketable innovations as quickly as possible. As part of this, Deutsche Telekom Laboratories focuses on five fields of innovation (5 i):

- Intuitive Usability of services and devices
- Integrated Service Components
- Intelligent Access
- Infrastructure for IT and telecommunications
- Inherent Security

The business and information systems engineering offers useful interdisciplinary approaches for all these areas of innovation. Subject matter includes, for example, modeling, methods and tools for process innovations, agile architectures for information and communication technologies (ICT), technology-oriented management approaches and techno-economic assessments. The aim is to safeguard the economic sustainability of innovations for the Group.

Deutsche Telekom Laboratories is divided into two areas: The Innovation Development Laboratory focuses on market-centric research and development within a timeframe of up to three years.

The basic and technology research of the Strategic Research Laboratory has a long-term focus. Common goal: Deutsche Telekom Laboratories is looking to become one of the world's leading research and development institutions in the field of new ICT.

An institute was set up together with Ben-Gurion University in Beer Sheva, Israel, in 2006. Since 2008, Deutsche Telekom Laboratories has also been represented in Darmstadt. Another project office was opened in the Silicon Valley, United States, in January 2009.

Contact:
Deutsche Telekom Laboratories
Ernst-Reuter-Platz 7
10587 Berlin, Germany
E-mail: wi.laboratories@telekom.de
www.laboratories.telekom.com

Deutsche Telekom Laboratories
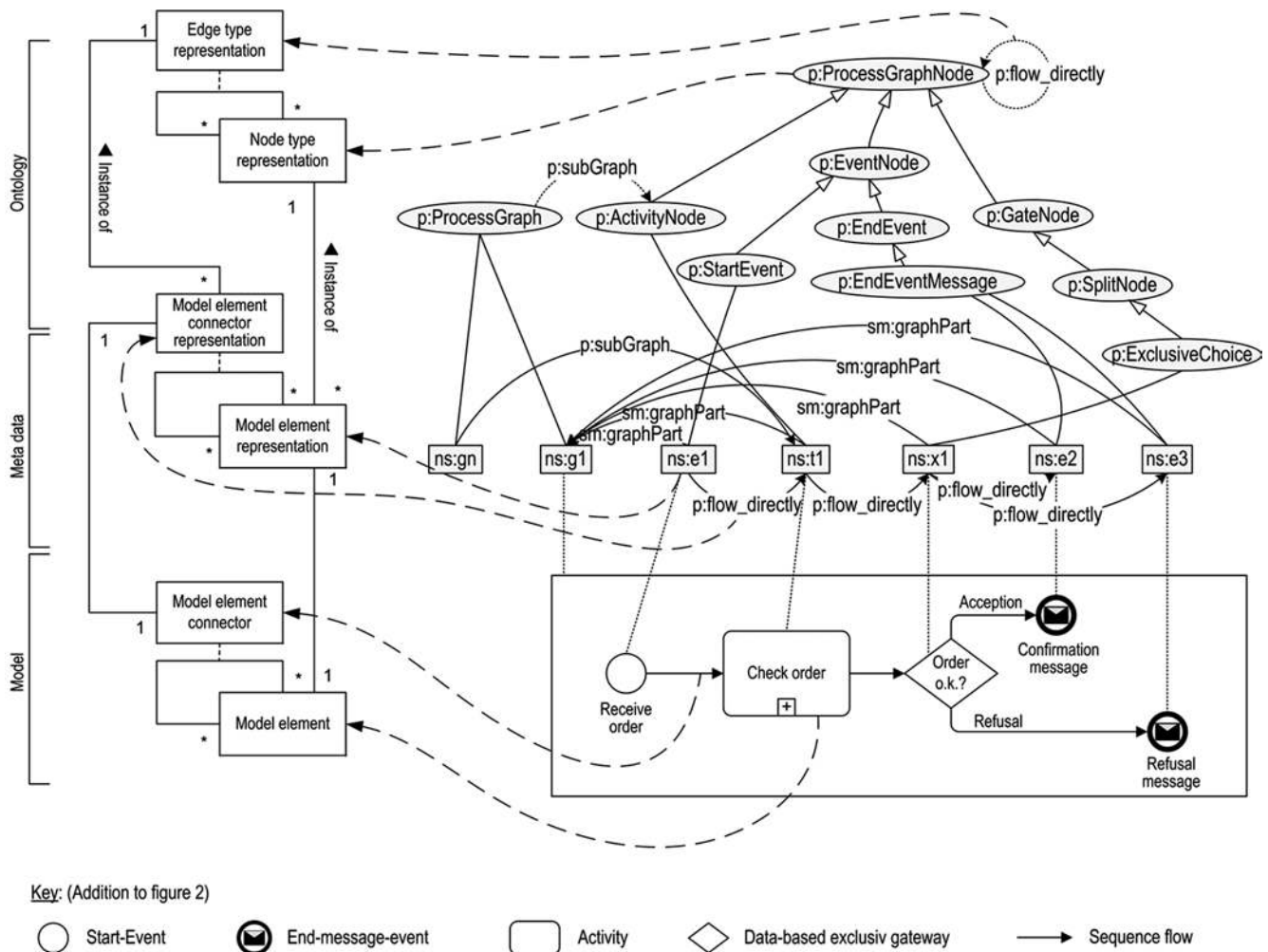An-Institut der Technischen Universität Berlin

Fig. 5 Transformation to BPMN language

a sub-process of y and supports the goal z, *then* this means that y also supports the goal z". With this rule the question "Which goals are supported by the activity *order_processing*?" can be answered with "*customer_satisfaction*". The reason for this is that the relation *supportsObjective*, which exists between the sub-activity *order_feedback* and the goal *customer_satisfaction*, is transferred to the superordinate activity *order_processing* (**Fig. 3**). This rule can be formulated in an informal syntax as follows:

```
Activity(?x)
∧ Activity(?y)
∧ subProcess _ directly(?x,?y)
∧ supportsObjective(?x,?z)
⇒ supportsObjective(?y,?z)
```

The IF-part of the rule is referred to as *antecedent* and describes a certain situation. The THEN-part is referred to as *consequent* and states a suitable conclusion, reaction or decision for the situation described in the IF-part.

Normative rules are applied to express conditions for the data used for an application or the logic used by it. They are also referred to as structural rules (Boley et al. 2007, p. 274) and can be further divided into consistency and integrity rules. Consistency rules are rules used to prevent contradictions in the ontology and the facts derived from it. This understanding of the term is related to terminology from the field of logic, where consistency refers to the property of an axiomatic system to not contain logical contradictions. An example for a consistency rule can be given with regard to the semantic annotation of process models. It should not be possible that – through a semantic annotation – a complete process model and a single individual element from that model are annotated with the same domain instance via the relation *equivalentTo*. In accordance with semantic integrity constraints from the database field, integrity rules are understood as rules used to maintain the

semantic correctness of the ontology and the facts derived from it. An example for an integrity rule can be given in regard to the validation of process models represented in the ontology against the background of **Fig. 3**. Thus for example, in a company a guideline regarding the improvement of customer communication could exist that states that after an order check is made, the customer should be informed about the result of the check.

There are a number of non-web-based ontology languages, such as OCML and Ontolingua, which make it possible to formulate rules without an extension. The ontology language OWL, used in this article, only supports the formulation of rules via extensions (apart from simple property chains in OWL 2.0). Such an extension is the Semantic Web Rule Language (SWRL) (Horrocks et al. 2004) which extends OWL with IF-THEN-rules in the form of a logical implication.

4.5 Generalization of the approach

Our goal in the model construction carried out here is to generalize the previously designed special process modeling, so that the knowledge won can be transferred to other modeling methods and languages, as well as to situations where semantically annotated models are being worked with. In contrast to the representation oriented towards examples in the former sections, we will now study the "building blocks" of semantic business processes and their relationships within the framework of a graph-theoretic interpretation. This applies to statements in which the elements of a *standard* business process, i. e. one without a special application, are collected. The class diagram from the Unified Modeling Language (UML) was selected as the modeling language.

Due to the graph-based interpretation of the process representations, the information model of the semantic process modeling is not the meta-model of a semantically extended modeling language or an ontology-based EPC, but rather a class model that specifies how business processes can be semantically annotated and represented independently of a certain process modeling language (**Fig. 4**). The information model for the semantic process modeling is divided into the three levels "Model", "Metadata" and "Ontology". The vertical layering of these levels in the data model does not, however, imply an abstraction relation. Instead it expresses the connecting function that the middle level, the metadata, has between the model on the one hand and the ontology on the other. Metadata refers here to data created by the annotation of a model with elements of the ontology.

On the model level, concrete models are represented by the classes *Model, Model element* and *Model element connector. Each Model element* (for example: "Create order" for an EPC-function or a BPMN-activity) is categorized by a Node type (for example: an EPC-function or BPMN-activity) and is assigned to at least one *Model* (for example: "Order entry"), which on its part is categorized by a *Model type* (For example: "EPC" or "BPMN"). It can also be related to other model elements via a *Model element connector* (for example the EPC-function "Create order" is followed by the EPC-event "Order has been created"), which is typified by an Edge type (for example an

Fig. 6 System architecture for the semantic process modeling
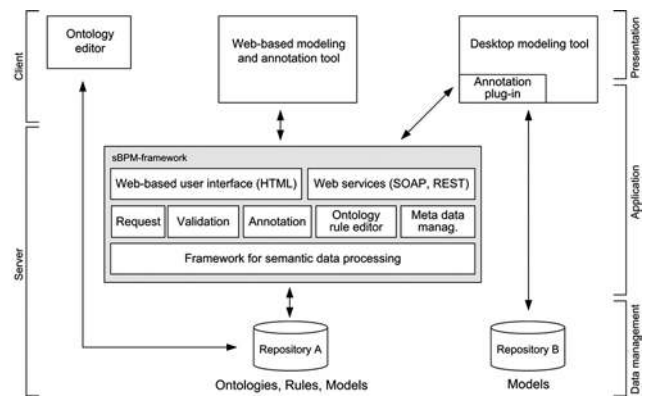
EPC-flow relation). The representation of the models on the metadata-level is conducted by means of transformation rules that state how elements on the model level can be transformed into elements on the metadata-level.

On the metadata-level models are represented by the classes *Model representation, Model element representation* and *Model element connector representation*, which are instantiated from the classes *Model type representation*, *Node type representation* and *Edge type representation* using the aforementioned transformation rules. A connection to the ontology level is created via the relationships with the association classes *Model annotation* and *Model element annotation*.

The central class at the ontology level is the *Ontology class*. It represents all classes contained in an ontology. Ontology classes can be related to one another. This circumstance is taken into account by the association class *Relation*. Because a relation between ontology classes can have a multitude of characteristics, such as for example a domain, range, value, label etc., it is introduced in the form of an association class. Ontology classes are divided into classes that represent constructs from a semi-formal language and those that represent the constructs of a domain. This circumstance is expressed by the specialization relations between the class *Ontology class* and the corresponding classes *Representation class* resp. *Domain class*.

Further rules can be formulated based on the ontology classes and instances. A *Rule* consists of an *Antecedent* and a *Consequent*. Both can not exist meaningfully outside of the rule containing them. This circumstance is made clear by a composition relation between the class Rule and the classes *Antecedent* and *Consequent*. In a similar manner, the *Antecedent* and the *Consequent* of a rule consist of an *Atom*

or several atoms, which also cannot exist meaningfully outside of their respective part of the rule. An *Atom* can refer to either a *Relation* or an *Ontology class*. This is represented by two alternative associations between which an {xor}-restriction exists. In addition, an *Atom* can refer to Ontology instances by using variables.

4.6 Transformation to other modeling languages

The shown information model is language neutral and it is possible to use it for other modeling languages. (**Fig. 5**) shows the idea using a BPMN (OMG 2006) example. The BPMN model, illustrated in the bottom area, is represented in the ontology above. Some parts of the information model from **Fig. 4** are on the left side. The directed and dashed edges, which go from the classes of the information model to the single elements on the right side, clarify the instance relations. Thus all classes contained in the ontology, which are used for the representation of process description language constructs, can be instantiated using the class *Node type representation* of the information model. The relationships between these classes can be instantiated using the class *Edge type representation*. The representation of model elements and their relations (connectors) in the ontology are instantiated by the classes *Model element representation* and *Model element connector representation*. The elements from the BPMN model are instances of the class *Model element* and *Model element connector*.

Compared to **Fig. 2**, only a few ontology extensions for the ontology-based representation of a BPMN model are necessary. Those refer to additional event types, for example the end event type "message", which is part of the BPMN language. The transformation from the shown approach
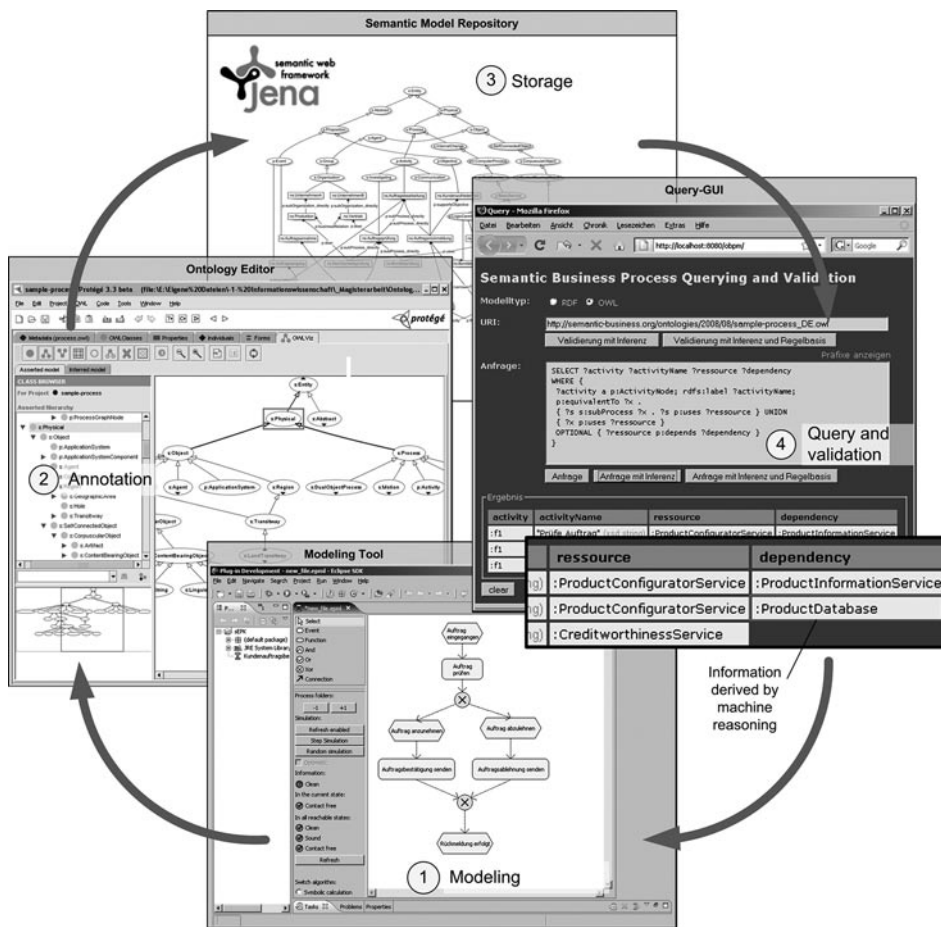
Fig. 7 Querying a semantically extended modeling tool

to other process modeling languages requires only a sub-class extension to reflect these different types of nodes.

As to the application of this approach, IT-support is particular important, referring to the application for this approach, especially for the modeling, annotation and machine processing of semantic, within the query and validation of process models. The following part presents the architecture and then the prototype implementation.

5 IT-support for semantic process modeling

5.1 System architecture

The system architecture shows the most important components and their cooperation for the IT-support of the semantic process modeling (**Fig. 6**).

An editor directly connected to the repository A, which saves ontologies, rules and models, is used to create and modify the ontology used for the annotation.

The center of the architecture is formed by a layered sBPM-framework (sBPM = semantic Business Process Management). A component for semantic data processing in the bottom layer provides basic functionalities for the components above it pertaining to the loading of data, its interpretation with an inference machine, and its modification and storage. The components above this layer provide the core functionalities needed for the IT-support of the semantic process modeling. These are the functionalities used by the modelers for the query, validation and annotation of process models during the construction and adjustment of models. The component mentioned last for annotation can provide the modeler with suggestions for the annotation on the basis of mappings of natural language on ontology terms (Niles and Pease 2003). Other elements of the sBPM-framework are components for metadata management, as well as an integrated ontology and rule editor. In contrast to the external ontology editor applied to create the ontology used, this editor is used to extend the ontology, whereby model constructors are given the

possibility of extending the ontology to fit their needs during the annotation procedure by, for example, deriving sub-classes or adding new instances.

Suitable interfaces for access must exist in order to use the functionalities via clients over a network. These are given in the superordinate layer in the form of a web interface (using the standards HTML resp. XHTML) and a web-service interface (using the standards SOAP/XML or REST).

When accessing the functionalities of the application level by a client, two scenarios are possible. First, access can take place via a web-based modeling and annotation tool (Scenario A). In this case, ontologies, rules and models can be stored together. Semi-formal models can thus be created, transformed and annotated with a single tool. The second possibility is that access occurs via a plug-in of a desktop modeling tool (Scenario B). In this case, the models can still be stored in the repository belonging to the respective modeling tool (Repository B), while their ontology-based representation is stored separately in Repository A. When using a desktop

modeling tool with a plug-in, only part of the functionalities of the semantic process modeling can be made available so that in the end two tools must be used.

5.2 Implementation of a test environment

The test environment sBPMQuery, based on the system architecture, was developed to demonstrate the applicability of the concept described above. It is composed of a modeling tool, an ontology editor and a server component with a web interface which allows query and validation of models. EPC Tools was used for the implementation as a process model editor, Protégé as an ontology editor, and Jena as a semantic web framework for the server components. In addition, the latter was extended by the inference machine Pellet. **Fig. 7** shows the relationships between the components based on the respective user interfaces.

Using the test environment an initial model (1) can be automatically transformed via a XSLT-script into its ontology-based representation, which is then imported into the ontology editor (2). After the model has been manually annotated in the ontology editor, the resulting semantic metadata is imported from the ontology editor to the storage in the server components (3). Communication with the server components is possible via a query interface (4), which runs as a client in a browser. With it, the models represented by the semantic metadata can be validated as well as queried with the query language SPARQL (Prud'hommeaux and Seaborne 2005). In addition, an inference machine or a rule execution component can be enabled for both functionalities. Thus, the interpretation of the data is possible on a semantic level, and facts contained in the model, but not explicit at the time of the query, can be inferred. Thus, the result of the example query on the right in **Fig. 7** is extended by the automatically inferred fact that the web service for the product configuration *ProductConfiguratorService* is dependent on the resource *ProductDatabase* (the English labels for the ontology terms in the context of the test environment slightly differ from the labels shown in **Fig. 3**). This can be inferred by the transitive relation depends, which exists – in the example in **Fig. 3**– between the web services *ProductConfiguratorService*, *ProductInformationService* and the database *ProductDatabase*. The knowledge thus derived can be used by the modeler for further modifications to the model (1).

6 Benefits of semantic process modeling

With the help of ontologies the semantics of process models can be represented in a formal way. This results in the following benefits for the approach of semantic process modeling developed in this paper:

- *Distributed modeling*: Semantic process modeling can increase the mutual understanding among the persons involved in the analysis and design of process-oriented information systems, as natural language labels for the modeling elements are annotated with the formally defined terms of a domain which have been modeled by an expert group. Especially for distributed modeling based on division of labor this may lead to a unified interpretation of the models.
- *Model management*: A model management that is based on semantic annotation of process models is of high benefit for the model management. It systematizes and facilitates access to the models and is suited for the support of the search and selection of process models. Here, key arguments are that, first, machine-processible semantics enhances the "fit" of a model in terms of the criteria of a query and, second, provides complete results for a search query, which does not require the user to explicitly specify every piece of information to receive an answer to a query due to automatic reasoning methods. This reduces the effort required to adapt the deployed reference models which are particularly offered by software houses and consulting firms.
- *IT-Business Alignment*: As part of the design and implementation of application systems, companies are confronted with the central challenge of having to take into account both technical and technological aspects. The problem here is that professional and technical modeling is usually not linked to each other. As a result of this existing gap a consistent transfer of technical requirements in supporting IT systems cannot be ensured. The necessary coordination between business and IT is highly simplified through the use of a jointly accepted vocabulary which is part of the semantic process modeling. Ultimately, this allows for achieving a higher business value contribution from IT in terms of IT-business alignment.
- *Compliance*: The use of rules that are based on an ontology open new possibilities for the validation of models. Inadequate models can be recognized at the time of modeling. Faulty representations may arise, for example, by the failure to comply with requirements which exist in the form of laws and regulations (e. g. Sarbanes-Oxley Act (SOX) for corporate reporting, ISO/IEC 27001:2005 for information security and DIN ISO 15489–1 for information and documentation) as well as in the form of internal regulations (compliance). The content-based validation which can be achieved by semantic process modeling can prevent the transmission of erroneous models to subsequent phases of modeling such as requirement specification.

7 Limitations and outlook

Seen from a conceptual point of view, the expenditure for the extension of modeling languages for supporting semantic technologies can be very high and may certainly overcompensate the benefits to be achieved by a model search or validation during modeling projects. In this regard, research must focus on evaluation questions in future, for example with the aid of criteria for the economic evaluation of the use of ontologies and semantically annotated models.

Moreover, we have to assess to what extent the use of the test environment sBPMQuery improves model development and application in business practice. It should be kept in mind that the central artifact of this contribution – the information model of semantic process modeling – is not the implementation model of the shown prototype. Therefore, the implementation of the test environment does not lead to any evaluation of the information model. Again, this may be the subject of further research.

In addition to a model evaluation, future research should focus on the embedding of semantic process modeling in existing organizational structures. Thus, it has to be clarified, for example, how semantic annotation can be organizationally

Abstract

Oliver Thomas, Michael Fellmann M.A.

Semantic Process Modeling – Design and Implementation of an Ontology-based Representation of Business Processes

An extension of process modeling languages is designed which allows representing the semantics of model element labels which are formulated in natural language by using concepts of a formal ontology. This combination of semiformal models with formal ontologies will be characterized as semantic process modeling. The approach is exemplarily applied to the languages EPC (Event-driven Process Chain), BPMN (Business Process Modeling Notation) and OWL (Web Ontology Language) and is generalized by means of an information model. The proposed formalization of the semantics of individual model elements in conjunction with the usage of inference engines allows the improvement of query functionalities in modeling tools and enables new possibilities of model validation. The integration of the approach in the IT-based work environments of modelers is demonstrated by a system architecture and a prototypical implementation. Evidently, advantages in the areas of modeling, model management, IT-business alignment, and compliance can be achieved by the application of modeling tools augmented with semantic technologies.

Keywords: Process management, Process modeling, Modeling languages, Semantics, Ontologies, Ontology languages

embedded in business practice, i. e. which employees perform the annotation, who maintains and adapts the ontology, and who synchronizes annotated models if elements in the business process or concepts in the ontology change.

References

Abramowicz W, Filipowska A, Kaczmarek M, Kaczmarek T (2007) Semantically enhanced business process modelling notation. In: Hepp M et al (eds) Proceedings of SBPM 2007. Innsbruck, pp 88–91

Ahlemann F, Teuteberg F, Brune G (2006) Ontologie-basierte Attributierung von Informationsmodellen: Grundlagen und Anwendungsgebiete. In: Teuteberg F, Ahlemann F (eds) ISPRI-Arbeitsbericht, Nr. 01/2006, Universität Osnabrück

AIIM (ed) (2007) BPM: Not just workflow anymore. Silver Spring

Bögl A, Schrefl M, Pomberger G, Weber N (2008) Semantic annotation of EPC models in engineering domains by employing semantic patterns. In: Cordeiro J, Filipe J (eds) Proceedings of ICEIS 2008. Barcelona, pp 106–115

Boley H, Kifer M, Patranjan PL, Polleres A (2007) Rule interchange on the web. In: Antoniou G et al (eds) Reasoning web. 3rd international summer school 2007. Springer, Heidelberg, pp 269–309

Bray T, Hollander D, Layman A, Tobin R (eds) (2006) Namespaces in XML 1.1. W3C

Brockmans S, Ehrig M, Koschmider A, Oberweis A, Studer R (2006) Semantic alignment of business processes. In: Manolopoulos Y et al (eds) Proceedings of ICEIS 2006. Paphos, pp 191–196

Cabral L, Domingue J, Motta E, Payne TR, Hakimpour F (2004) Approaches to semantic web services: An overview and comparisons. In: Bussler C et al (eds) The semantic web: Research and applications: Proceedings of ESWS 2004. Springer, Berlin, pp 225–239

Cardoso J, Sheth AP (2005) Introduction to semantic web services and web process composition. In: Semantic web services and web process composition. 1st International Workshop, SWSWPC 2004, San Diego, pp 1–13

Casely-Hayford L (2005) A comparative analysis of methodologies, tools and languages used for building ontologies. CCLRC, Swindon

Desel J (2002) Model validation – A theoretical issue? In: Esparza J, Lakos C (eds) Application and theory of Petri nets. Springer, Heidelberg, pp 23–43

Dietz JLG (2006) The deep structure of business processes. Comm. ACM 49(5):58–64

Fensel D, Hendler J, Lieberman H, Wahlster W (eds) (2003) Spinning the semantic web. MIT Press, Cambridge

Fox MS (1992) The TOVE project: A common-sense model of the enterprise. In: Belli F, Radermacher FJ (eds) Industrial and engineering applications of artificial intelligence and expert systems. Springer, London, pp 25–34

Gómez-Pérez A, Fernández-López M, Corcho O (2004) Ontological engineering. Springer, London

Gómez-Pérez A, Manzano-Macho D (eds) (2003) A survey of ontology learning methods and techniques. Universidad Politécnica de Madrid

Green PF (1996) An ontological analysis of information systems analysis and design (ISAD) grammars in upper case tools. PhD Thesis, University of Queensland, Brisbane

Green PF, Rosemann M (2000) Integrated process modeling: An ontological evaluation. Information Systems 25(2):73–87

Green PF, Rosemann M, Indulska M (2005) Ontological evaluation of enterprise systems interoperability using ebXML. IEEE Transactions on Knowledge and Data Engineering 17(5):713–725

Gruber TR (1993) A translation approach to portable ontology specifications. Knowledge Acquisition 5(2):199–220

Hahn A (2005) Integration verteilter Produktmodelle durch Semantic-Web-Technologien. WIRTSCHAFTSINFORMATIK 47(4):278–284

Heinrich B, Bewernik M-A, Henneberger M, Krammer A, Lautenbacher F (2008) SEMPA – Ein Ansatz des Semantischen Prozessmanagements zur Planung von Prozessmodellen. WIRTSCHAFTSINFORMATIK 50(6):445–460

Henneberger M, Heinrich B, Lautenbacher F, Bauer B (2008) Semantic-based planning of process models. In: Bichler M et al (eds) Multikonferenz Wirtschaftsinformatik 2008. GITO, Berlin, pp 1677–1689

Hepp M (2005) eClassOWL: A fully-fledged products and services ontology in OWL. In: Poster Proceedings of ISWC2005. Galway

Hepp M, Leymann F, Domingue J, Wahler A, Fensel D (2005) Semantic business process management: A vision towards using semantic web services for business process management. In: Proceedings of IEEE ICEBE 2005. Beijing, pp 535–540

Hepp M, Roman D (2007) An ontology framework for semantic business process management. In: Oberweis A et al (eds) eOrganisation: Service-, Prozess-, Market-Engineering: 8. Internationale Tagung Wirtschaftsinformatik, vol 1. Universitäts-Verlag, Karlsruhe, pp 423–440

Horrocks I, Patel-Schneider PF, Boley H, Tabet S, Grosof B, Dean M (2004) SWRL: A semantic web rule language. W3C

Keller G, Nüttgens M, Scheer A-W (1992) Semantische Prozeßmodellierung auf der Grundlage "Ereignisgesteuerter Prozeßketten (EPK)". In: Scheer AW (ed) Veröffentlichungen des Instituts für Wirtschaftsinformatik, Nr. 89. Universität des Saarlandes, Saarbrücken

Kindler E (2006) On the semantics of EPCs: Resolving the vicious circle. Data & Knowledge Engineering 56(1):23–40

Koschmider A, Ried D (2005) Semantische Annotation von Petri-Netzen. In: Workshop für Algorithmen und Werkzeuge für Petrinetze (AW-PN'05). Humboldt-Universität Berlin, pp 66–71

Kuropka D (2004) Modelle zur Repräsentation natürlichsprachlicher Dokumente. Logos, Berlin

Lautenbacher F, Bauer B (2006) Semantic reference- and business process modeling enables an automatic synthesis. In: Hinkelmann K et al (eds) Proceedings of SBPM 2006. Budva, pp 89–100

Lin Y (2008) Semantic annotation for process models: Facilitating process knowledge management via semantic interoperability. PhD Thesis, Norwegian University of Science and Technology, Trondheim

Lin Y, Ding H (2005) Ontology-based semantic annotation for semantic interoperability of process models. In: Proceedings of CIMCA-IAWTIC '06, vol 1. IEEE, Washington, DC, pp 162–167

Lin Y, Strasunskas D (2005) Ontology-based semantic annotation of process templates for reuse. In: Proceedings of 10th CAiSE/IFIP8.1/EUNO international workshop on evaluation of modeling methods in system analysis and design (EMMSAD05). Porto, pp 162–167

Miles A, Bechhofer S (2008) SKOS simple knowledge organization system reference. W3C

Niles I, Pease A (2001) Towards a standard upper ontology. In: Welty C, Smyth B (eds) Proceedings of the 2nd international conference on formal ontology in information systems (FOIS-2001). Ogunquit, 2001, pp 2–9

Niles I, Pease A (2003) Linking lexicons and ontologies: Mapping WordNet to the suggested upper merged ontology. In: Proceedings of the 2003 international conference on information and knowledge engineering (IKE 03). Las Vegas, pp 23–26

OMG (ed) (2006) Business process modeling notation specification. OMG, Needham

Ogden CK, Richards IA (1923) The meaning of meaning, 1st edn. Kegan Paul, London

Ortner E (1997) Methodenneutraler Fachentwurf. Teubner, Stuttgart

Prud'hommeaux E, Seaborne A (eds) (2005) SPARQL query language for RDF. W3C

Rosemann M (1996) Komplexitätsmanagement in Prozessmodellen. Gabler, Wiesbaden

Rosemann M, Schwegmann A (2002) Vorbereitung der Prozessmodellierung. In: Becker J et al (eds) Prozessmanagement, 3rd edn. Springer, Heidelberg, pp 47–94

Smith MK, Welty C, McGuiness DL (eds) (2004) OWL web ontology language guide. W3C

Soffer P, Golany B, Dori D, Wand Y (2001) Modelling off-the-shelf information systems requirements: An ontological approach. Requirements Engineering 6(3):183–199

Studer R, Benjamins VR, Fensel D (1998) Knowledge engineering: Principles and methods. Data & Knowledge Engineering 25(1–2):161–197

Thomas O, Fellmann M (2007) Semantic business process management: Ontology-based process modeling using event-driven process chains. International Journal of Interoperability in Business Information Systems 2(1):29–44

Uschold M, King M, Moralee S, Zorgios Y (1998) The enterprise ontology. The Knowledge Engineering Review 13(1):31–89

van der Aalst WMP, ter Hofstede AHM, Kiepuszewski B, Barros AP (2003) Workflow patterns. Distributed and Parallel Databases 14(1):5–51

Wand Y, Weber R (1995) On the deep structure of information systems. Information Systems Journal 5(3):203–223

Zelewski S, Alan Y, Alparslan A, Dittmann L, Weichelt T (eds) (2005) Ontologiebasierte Kompetenzmanagementsysteme. Logos, Berlin