ECIS 2007 Proceedings

European Conference on Information Systems (ECIS)

2007

Achieving Business Process Model Interoperability Using Metamodels and Ontologies

P Hofferer
University of Vienna, ph@dke.univie.ac.at

ACHIEVING BUSINESS PROCESS MODEL INTEROPERABILITY USING METAMODELS AND ONTOLOGIES

Höfferer, Peter, Department of Knowledge and Business Engineering, University of Vienna, Brünner Straße 72, 1210 Vienna, Austria, ph@dke.univie.ac.at

Abstract

Common user- and business-requirements demand an integrated view on the multitude of available information resources that are processed in a variety of heterogeneous information systems. Therefore, the research field of semantic integration which is a prerequisite for the interoperability of systems is of high importance. In this paper we focus on the semantic interoperability of business process models which is, for instance, of relevance for the interoperability of standard software or workflow engines that are configured by such processes. In order to perform such a task, the use of two alternative concepts is often proposed in literature: metamodels and ontologies. These two concepts are most often used without really reflecting their characteristics and their relationship to one another which is a critical shortcoming. Therefore, we provide an extensive discussion of these concepts which establishes the basis for their combined use for achieving semantic interoperability of business processes. In this context we will also discuss basic approaches for the mapping of model elements and ontology constructs which is of importance for providing inherent semantics. *The paper is concluded with an outlook on the ongoing implementation of the proposed interoperability approach and its possible benefits for the area of service oriented architectures.*

Keywords: semantic integration, interoperability, metamodels, ontologies, business process modeling.

1 INTRODUCTION

The topic of integrating data and ensuring the interoperability of information systems is of great practical importance which can already be seen by the fact that according to Gartner up to 40% of the companies' information technology budgets are spent on integration issues (Haller et al. (2005)). The heterogeneities that have to be dealt with in this context are usually classified to be of syntactical, structural or semantic nature (Obrst (2003)) whereas resolving the latter seems to be most laborious as 60-80% of the resources of integration projects are spent on reconciling semantic heterogeneities (Doan, Noy and Halevy (2004)).

In the paper at hand we basically focus on the interoperability of business processes. In order to achieve such a goal we need an adequate representation of these processes which is provided by a variety of particular modeling methods like Event-Driven Process Chains (EPC; Keller, Nüttgens and Scheer (1992)), the Business Process Modeling Notation (BPMN; Object Management Group (2006a)), LOVEM (IBM (2006)), IDEF 3 (Knowledge Based Systems Inc. (1995)), or BPMS (ADONIS; Karagiannis, Junginger and Strobl (1996)). These methods provide means to generate diagrammatic models (Harel and Rumpe (2000)) that have to be mapped with each other in order to provide interoperability. In this context it has to be taken into account that often not only a mapping between models that have been created using the same business process modeling method has to be done, but also between models that comply to different ones.

The task of the semantic integration of business process models which implies the identification of model elements that carry the same meaning is a *"formidable problem"* as a direct mapping between such elements can hardly be established (Zaniolo and Melkanoff (1982)). Thus, other means have to be applied to support the execution of this task. Basically, there are two different approaches: on the one hand *metamodels* (e.g. Bézivin (2005)) are used and on the other hand *ontologies* (e.g. Jamadhvaja and Senivongse (2005)).

These two concepts are often used without giving a clear definition of what they actually are. This is a shortcoming that also implies that the functionality that is provided by these concepts is not used to the full extent. In order to make up for this we will discuss the nature of metamodels and ontologies in the following section 2 that will be closed with a critical comparison. This provides us with the insight that a combined use of both approaches is a prerequisite of enhanced semantic interoperability of business processes which we will discuss in section 3. We will show there that a full semantic description of model elements can only be achieved via both metamodels that describe *type semantics* and ontologies that define *inherent semantics*. While the relationship between model elements and their metamodel is implicitly defined, this is not the case for the ontology. Therefore, we will also discuss some basic approaches for establishing the links between model elements and ontology concepts. The paper is concluded in section 4 giving an outlook on future conceptional work, the ongoing prototypical implementation of the proposed interoperability approach, and possible benefits for applications in the area of service oriented architectures.

2 BASIC CONCEPTS

This section is giving a brief overview on the fields of both metamodels and ontologies concluding with an analysis of the relationship between these two concepts. This critical comparison will provide the basis for the further discussion of the combination of these concepts in order to realize interoperability of models.

2.1 Metamodels and Business Process Modeling

Before talking about metamodels we want to introduce the notion of a *modeling technique* that allows us to define the necessary components for the creation of models. According to Kühn (2004) a modeling technique describes the modeling constructs of a *modeling language* (i.e. entities, relationships, attributes, activities, and the like) and a *modeling procedure* that defines how these constructs have to be used in order to create a valid model (see Figure 1).

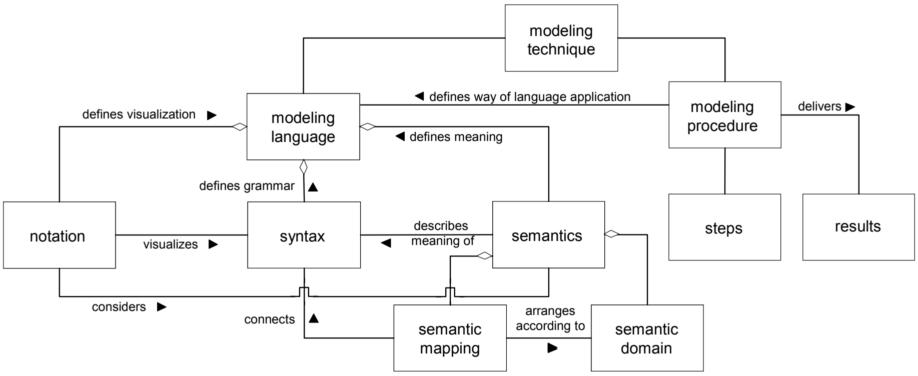

Figure 1. Elements of a modeling technique. (Kühn (2004))

Following Harel and Rumpe (2000) a modeling language now consists of syntax which focuses *"purely on notational aspects"* and semantics which defines the meaning. Kühn (2004) extends this view in that he separates notation from syntax as he defines notation as the *"representation of the elements of the language"*. Syntax then is how the representation elements are allowed to be combined. We consider this distinction as important as it allows for changing only syntax or only notation in the context of method engineering without affecting the other.

Now we can talk about metamodels. Often they are simply referred to as being just *"models of models"* (Object Management Group (2003), p. A-2) which is basically correct but can be quite misleading. The plural is of importance here as a metamodel is always the description of a set of other models in that *"A metamodel is a model of a modelling language."* (Favre (2005)) Therefore, they describe the language constructs that can be used for the creation of models (Klint, Lämmel and Verhoef (2005)). It is important to stress that a metamodel is not just "a model of a model" (please notice the singular) (Favre (2005), Bézivin (2005)).

Therefore, the task of creating a metamodel is the task of creating a language that is capable to describe the relevant aspects of a subject under consideration that are of interest for the future users of the created models. The relationship between models and metamodels shall be visualized in a *"metamodeling hierarchy"* in the following Figure 2 in order to enable a better understanding.

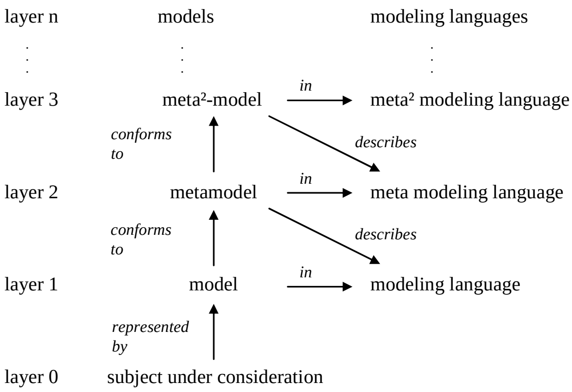

Figure 2. The metamodeling hierarchy. (adapted from Strahringer (1996))

It can be seen here that subjects under consideration are *represented by* models. These are created using a modeling language that is *described* by a metamodel which implies that a model *conforms to* a metamodel. We have decided to use these italicized terms according to Bézivin and Favre[1]. We can further see in Figure 2 that metamodels themselves can be created using another (meta) modeling language that is *described* by a meta²-model (also sometimes referred to as meta-metamodel). Theoretically, this chain of metamodels can be carried on to the n-th level. Practically, the process of creating metamodels is most often stopped at the meta²-level whereas the description of the meta² modeling language is reflexive. This can for example be compared with the self-representation of the EBNF notation that can be realized with a few statements in EBNF (Bézivin (2005)).

Concluding, we want to show in Figure 3 how a simplified metamodel and a corresponding model instance in the area of business process modeling could look like.

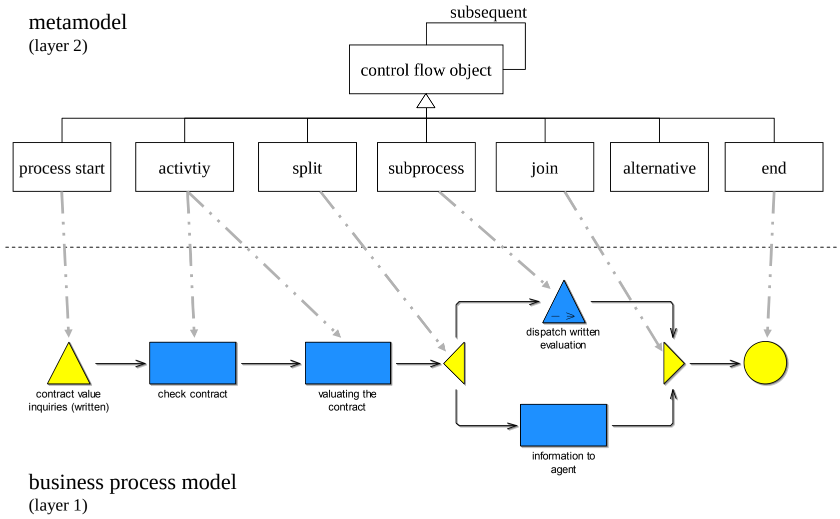

Figure 3. The relation between a metamodel and a business process model.

2.2 Ontologies and Business Process Modeling

The term ontology originally stems from the field of philosophy where it is *"the science of what is, of the kinds and structures of objects, properties, events, processes and relations in every area of reality"* (Smith and Weltym (2001)). Therefore, the most basic objective of ontology is simply to describe reality.

Ontologies are applied in a wide range of computer science-related research fields (Guarino (1998), McGuinness (2003)), but the type of the actual work that is done using this generic term is varying significantly (Green and Rosemann (2005)). Almost all approaches refer to a definition provided by Gruber (1993) which is *"An ontology is an explicit specification of a conceptualization."* Conceptualization hereby is to be understood as *"an abstract, simplified view of the world that we wish to represent for some purpose."* Gruber's original definition is often extended by two words that he also discusses in his paper[2]: the specification shall be "formal" which means that it can be processed by machines and the conceptualization shall be a "shared" consensus within a certain community that is about to use the ontology.

[1] Both Bézivin and Favre argue that the nomenclature *conforms to* is to be preferred over *instance of* which is for example used by Atkinson and Kühne (2002) in the same context. For a detailed discussion of this please refer to Bézivin (2005) and Favre (2005).

[2] The extended version of Gruber's original definition can be found in Studer et al. (2000) or Uschold and Gruninger (2004), for instance.

But as Smith and Weltym (2001) emphasize, Gruber's definition is quite generic and *"leaves room for too many possible interpretations"*. Therefore, we want to refer to some more specific definitions as well:

- An ontology is considered *"as a set of logical axioms designed to account for the intended meaning of a vocabulary."* (Guarino (1998))
- An ontology is an *"engineering artifact* [which is] *constituted by a specific vocabulary used to describe a certain reality, plus a set of explicit assumptions regarding the intended meaning of the vocabulary words."* (Guarino (1998))
- *"[…] an ontology is a document or file that formally defines the relations among terms. The most typical kind of ontology for the Web has a taxonomy and a set of inference rules."* (Berners-Lee, Hendler and Lassila (2001))

From all these comments we can derive our definition of ontologies that we will use throughout our paper: "Ontologies define vocabularies that have to be shared by many people in an explicit and machine-processable way." Such vocabularies now can have varying detail with respect to the defined relationships of their words. Obrst (2003) is defining a so-called "ontology spectrum"[3] that is reflecting this fact. This spectrum distinguishes between ontologies that contain "weak semantics" and "strong semantics". Controlled lists of words are the simplest form of an ontology. Taxonomies in the next step introduce super- and sub-relationships within this controlled vocabulary. Thesauri then provide linguistic relationships for defining homonyms, synonyms, and the like. Finally, topic maps allow for the definition of arbitrary additional types of relationships. In our view all these concepts are ontologies whereas some are more powerful with respect to their semantic expressiveness than others. We are aware of the fact that there is no consensus in the community concerning this. This shall be illustrated via the following citation from McGuinness (2003): *"Some people consider the previous categories (of catalogs, glossaries, and thesauruses) to be ontologies, but many prefer to require that an explicit hierarchy [is] included before something is considered an ontology."*

In the context of business process modeling, ontologies are - according to our definition – a means to provide the vocabulary of an application domain that is then used for the creation of process models. So an ontology could describe all existing concepts within the banking or insurance area, for instance. Hereby, the vocabulary can be based on commonly agreed industry standards or can also be an organization-specific one.

Now, we want to use our comments on metamodels and ontologies as basis for the following critical comparison.

2.3 Metamodels and Ontologies: A Synergy Effect

It is quite interesting to see that there is no commonly agreed view on the relationship between metamodels and ontologies in the scientific community. In order to illustrate this we want to give a few examples in the following:

- In Kayed and Colomb (2005) it is stated that a *"meta-model restricts the way of defining terms […] to represent ontological concepts"*.
- *"[…] the joint use of metamodeling and ontologies allows to describe domain knowledge for a complex domain. Ontologies are used as stabilized descriptions of a business domain while metamodels allow a fine description. […] metamodels and ontologies present common "deep" characteristics."* (Terrasse et al. (2006))
- An *"ontology can be regarded as a special kind of metamodel"*. (He, He and Wang (2005))
- *"[…] the ontology explicitly expresses the semantics of the modeling concepts whose syntax is defined by the metamodel."* (Kramler et al. (2006))

[3] Similar ontology classifications are provided by McGuinness (2003) and Smith and Weltym (2001).

As we can see here ontologies and metamodels are sometimes considered to be of the same nature (He, He and Wang (2005)) whereas at other times there is a clear distinction between them. One view here is that metamodels provide the means to create ontologies (Kayed and Colomb (2005)). We agree with that as ontologies are, of course, also models and, therefore, need to use a language that is defined by metamodels. There is plenty of work available that is dealing with the elaboration of metamodels for ontologies like Object Management Group (2006b).

Our main argument concerning the relationship between metamodels and ontologies is a different one. We see metamodels as provider of the syntax of a modeling language which means that all available modeling constructs are defined as well as valid ways to combine them. Semantics has only a negligible role here. Of course, there is always some semantics included in language constructs like "entity" or "attribute" but only in an implicit (i.e. not directly machine-processable) manner. Ontologies on the other hand basically provide the semantics and they have a twofold role in this context: they can describe both the semantics of the modeling language constructs (Kramler et al. (2006)) as well as the semantics of model instances.

In the context of business processes this means that ontologies can be used for the explication of both the semantics of activities in general as well as for describing the semantics of a specific activity in that, for instance, the used resources or the application context are described by a domain ontology. Therefore, in our view ontologies fulfill a cross-sectional function (also see Figure 4) whereas semantic interoperability can only be achieved when both concepts – metamodels and ontologies – are used in combination.

3 BUSINESS PROCESS MODEL INTEROPERABILITY: A SEMANTIC APPROACH

As has been shown, metamodels and ontologies are different but complementary concepts. To our knowledge Atkinson and Kühne (2003) have been one of the first who explicitly realized this when talking about "linguistic -" and "ontological metamodeling". Before we present our approach for achieving semantic interoperability of business processes we will briefly refer to some related work in the following.

- Kappel et al. (2006) as well as Roser and Bauer (2005) are working in the area of model driven (software) development (MDA, Object Management Group (2003)). They want to enable the automatic transformation of models in order to be able to process them within different commercial CASE tools that make use of different model formats. In order to do so they first define metamodels for all tools that are then each linked to one tool ontology. These ontologies are bound to one single, superior "generic ontology" that is then used to find correspondences between the tool ontologies. The mappings of the tool ontologies are in the next step transferred to "bridgings" between the metamodels which finally enables the transformation of CASE-model instances.
- Pan and Horrocks (2001) also use a combination of metamodels and ontologies in order to "annotate" model instances but in their approach the ontology concepts are created corresponding to the same metamodel that is also used to create model instances. We see this is a restriction that is not necessary.
- We have already referred to Terrasse et al. (2006) in section 2.3. They make – in their own words – use of ontologies for early categorization and metamodels for a fine description in order to describe domain knowledge. Integration per se is no topic there. Although the two proposed steps seem to be a bit vague, one very interesting use case can be found there: the ontology is used to restrain the allowed values for attributes of model instances.

Now it is time to turn to our view of the combined use of metamodels and ontologies for achieving semantic interoperability of business processes. The conceptional architecture is as shown in Figure 4. Different models of the bottom model-layer are created corresponding to different metamodels that in turn are created using one common meta^2-model[4]. With the help of this "linguistic metamodeling" primarily syntactical but also some semantic aspects of model elements are defined. We will discuss this in more detail later on in an example scenario.

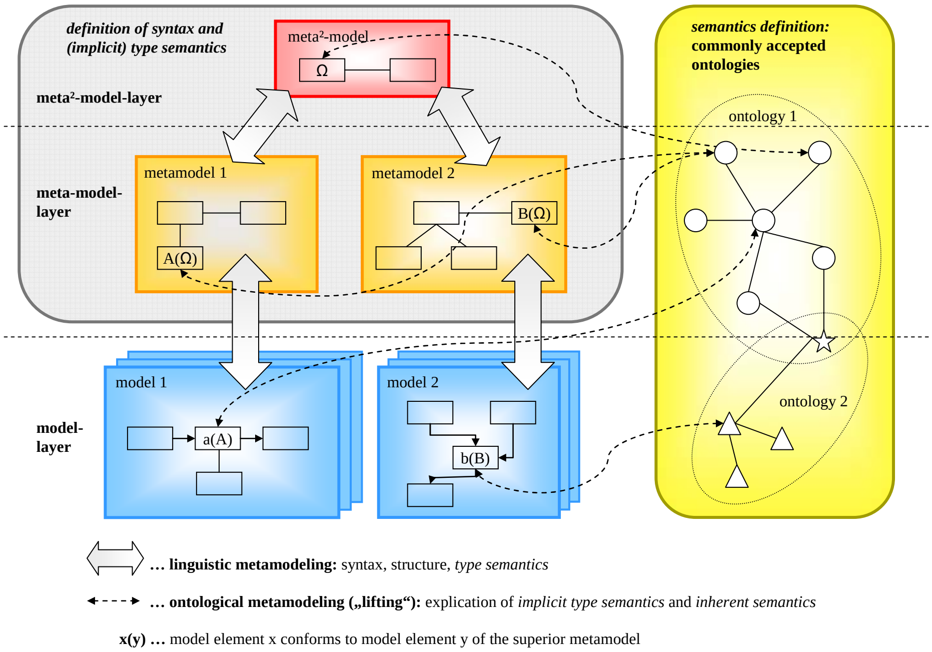

Figure 4. Architecture for semantic interoperability using both metamodels and ontologies.

As opposed to Pan and Horrocks (2001) we see ontologies completely independent from the language concepts that are used to create the models that are to be integrated[5]. It is not necessarily the case that one single ontology will be sufficient to integrate all models. We have indicated this in Figure 4 where we have depicted two different ontologies that are interrelated to each other via one common concept that is visualized as a star.

The basis for semantic interoperability is provided via linking model elements of arbitrary layers of the metamodel hierarchy with ontology concepts. This process is known as *lifting* (Kappel et al. (2006)) or *ontology anchoring* (Brinkkemper, Saeki and Harmsen (1999)) and fulfills the task that Atkinson and Kühne (2003) call ontological metamodeling. We will discuss some possible approaches for this later on. First we will demonstrate in an example how the semantic integration of two business process models is accomplished using our architecture.

Let's consider a simple scenario of a merger of two companies of the same business sector. Such a merger usually implies the consolidation and optimization of existing business processes within these two companies in order to identify and utilize existing synergies. A prerequisite for this is the comparison of existing business process models with the objective of finding semantically equivalent procedures.

Basically, such semantic integration has to be realized by two different, complementary means. First, models have to be compared and integrated with respect to the description of their language. In our

[4] Theoretically, the common metamodel could also be defined not until the n-th layer but most often this is done at the second metalayer (cf. e.g. MDA Object Management Group (2003)).
[5] But of course, the ontologies have to comply with their own, independent metamodel.

example this means that only model elements of the same type are compared with each other. Activities should be compared with activities, events with events and not, for instance, activities with events as these two language elements do refer to completely different things: an activity describes a dynamic procedure where something is done by some actor using some resources whereas an event is something that just happens at a discrete point of time. Metaphorically speaking, it must be ensured that one is not comparing "apples and oranges". This functionality can be provided by the metamodels of the modeling languages that define what we would like to denote as *type semantics* (cf. Karagiannis and Höfferer (2006) and Figure 4). Metamodel- and therefore language constructs are used as a filter for a first comparison of similarity (also cf. Bézivin (2005)). We would like to call this *metamodel-based integration*.

While such metamodel-based integration is quite simple in the case of comparing models that have been created using the same language, it is a powerful means if models are to be integrated that conform to different metamodels. In our scenario, for instance, it is most likely that the two companies are using different modeling languages in order to describe their business: Company X might use EPC whereas company Y utilizes BPMS. Although, these languages are very similar as they both aim for describing business processes they are not identical. While both contain metamodel classes for modeling "activities" and the same operators for defining the control flow (even though, they might be named differently), the metamodel element "transaction" in BPMN does not have any direct equivalent in BPMS or EPCs, for instance. Therefore, in order to guarantee the correct integration of business process models it has to be ensured first that there is a correct mapping of their modeling languages available. In our case it has to be made explicit, for instance, that "BPMS-activity" is equivalent to "EPC-activity" and is not to be compared to "EPC-event". So we see here that the integration task of models is passed up in the metamodeling hierarchy and also implies the integration of metamodels. In Karagiannis and Höfferer (2006) it is stated that the integration of different metamodels always depends on the existence of a common meta²-model[6]. Related work concerning the creation of such *integrated metamodels* (i.e. integrated modeling languages) can, for example, be found in Kühn and Murzek (2006). Integrated metamodels can also be used to perform translations from one modeling language into another. Thus, it is possible to transform an EPC business process model into a BPMS model.

Metamodel-based integration can be enhanced by using techniques stemming from computer linguistics and text retrieval. Via the use of more or less elaborated algorithms for string comparison a step beyond simple type semantics can be made. Thereby, it is possible, for instance, to determine that activities which descriptions contain the words "proposal" or "proposals" are not only related because of just being activities but, furthermore, because they are operating on the same resource. The problem is that such an approach has a lack of real understanding of the meaning of the strings that are compared. This can lead to severe mismatching in the case of homonyms as, for example, "bank" can mean shore in a geographical context as well as financial institution in another context. A simple string comparison is not able to recognize this different semantics. On the other hand synonyms can not be recognized which is a shortcoming that is of extreme relevance in our scenario as different companies are most likely to use different nomenclature in order to denominate the same concepts of reality. For demonstration purposes a "proposal" in company X might be a "suggestion" in company Y.

Such equivalence can only be found and used if the *inherent semantics* of model content (cf. Karagiannis and Höfferer (2006)) is taken into account. This is the meaning that is created when constructs of the modeling language (i.e. metamodel) are used to model concrete real world phenomena. Creating an activity "refuse proposal", for instance, describes a process of not accepting

[6] Sometimes this common meta²-model is not explicitly available. In these cases the lifting of metamodel elements (layer 2 in Figure 2) to ontologies can allow for their semantic integration in the same ways that we are about to discuss later on in this section for the integration of models (layer 1).

what somebody has proposed to do. But of course, we can not infer this from the simple fact that "refuse proposal" is an activity. In order to do this we need more information than that which can be provided using some of the facilities that have generally been classified as ontologies (cf. section 2.2). Elements of different models that are now linked to the same ontology concept can be considered to be equivalent or at least closely related considering their inherent semantics. The integration that can be realized using this approach shall be referred to as being *ontology-based* further on. In order to create an appropriate ontology for this purpose, existing domain ontologies can be used as well as research results in the field of ontology mapping – see Kalfoglou and Schorlemmer (2005) or Noy (2004).

Now we want to demonstrate how ontology-based combined with metamodel-based integration allows us to recognize that, for instance, an activity "refuse proposal" in an EPC business process model is equivalent to an activity "reject suggestion" in a BPMS model (also see Figure 5):

- First of all, through metamodel- and therefore modeling language integration we have generated an integrated metamodel that contains the information that "EPC-activities" are equivalent to "BPMS-activities".
- In the next step we are able to recognize synonyms, i.e. different words that refer to the same semantic concept. "Proposal" and "suggestion", for instance, are both something that is put forward for consideration or acceptance through an authority.
- According to the structure of our underlying ontology that includes a relationship "used together with" we are also able to recognize that both a "suggestion" as well as a "proposal" can be "refused" or "rejected". This relationship can be generally used to define which verbs can be used together with which nouns. This way a controlled vocabulary can be provided for the creation of new business process models that might define that proposals can be accepted or refused.

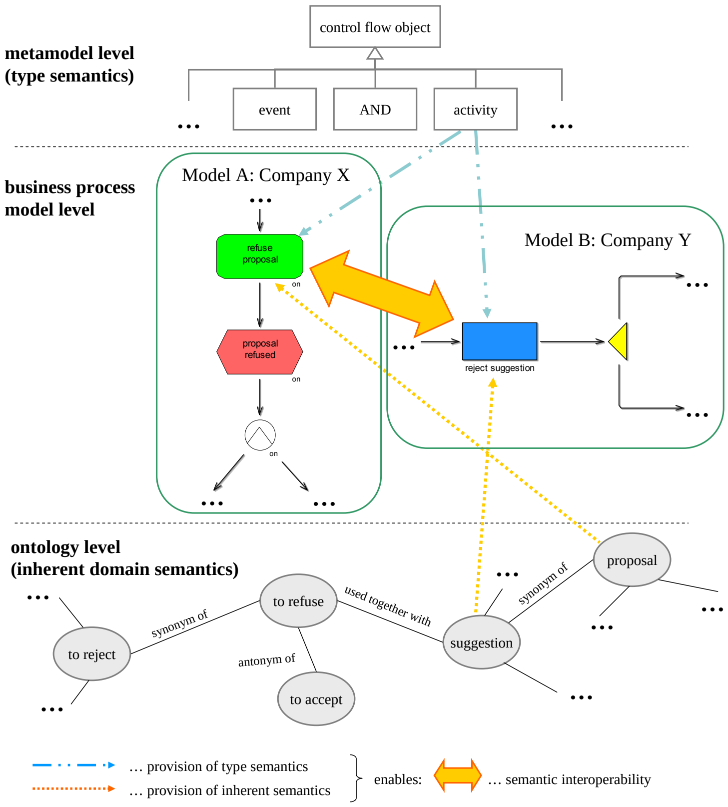

Figure 5. Semantically interoperable business process models.

So we see here that only a combination of metamodel type semantics together with inherent semantics from ontologies provides a means for semantically interoperable business process models. Hereby, our example has only demonstrated some of the advantages that we can benefit from when using ontologies. Through an "antonym of" relation we could also identify activities that do exactly contrary things. If the ontology provides us with inheritance (i.e. super- and subclass relationships) between concepts, we can make use of this hierarchical information as well. Finally, also far more complex rules can be defined between ontology concepts that can reflect domain specific characteristics and further enhance the quality of semantic interoperability.

Thus, the relationship between model elements and ontology concepts has a crucial role in our architecture. Thereby, the ontologies provide an external knowledge base that is basically completely independent from the metamodels and models and the links are established following a denotational approach for the explication of semantics (associative coupling). Now the question arises how to actually establish these links between the (meta-)model elements M and the ontology concepts O ($M \leftrightarrow O$ for short, which represents the “ontological metamodeling” arcs in Figure 4). The following approaches can be considered:

- *Manual approach.* The manual creation of $M \leftrightarrow O$ is most likely to provide the best results given that skilled domain experts are engaged in this task. But on the other hand, manual lifting can only be done if a certain amount and size of metamodels and ontologies is not exceeded. If this is the case some type of automation has to be applied.
- *Adaptive approach.* One possibility for achieving automation is the use of neural networks that are characterized by their capability to learn by examples. Such a neural network can be trained with the help of some $M \leftrightarrow O$ mappings that have been created manually in order to be able to predict fitting ontology concepts for arbitrary, previously unknown (meta-)model elements.
- *Computational approach.* As the $M \leftrightarrow O$ links basically rely on the comparison of textual strings it seems to be reasonable to take into account possibilities provided by computer linguistics. Here a variety of similarity measures exist in the form of

  $$sim : (M_i) \times (O_j) \rightarrow [0...1].$$

  Such measures can reflect both syntactic as well as semantic nearness of strings (see Ehrig et al. (2005) or Pedersen, Patwardhan and Michelizzi (2004)). For example, it is possible to state here that “proposal” and “suggestion” are similar to a degree of 0.9333 whereas “proposal” and “car” are only similar to a degree of 0.2353[7]. Of course, extensive evaluation work concerning the quality of different measures has to be done in order to ensure good results.

In the end a combined semi-automatic approach for the lifting of (meta-)model elements seems to be the most promising solution. Automatic mechanisms will be used for generating mapping recommendations that have to be approved by human experts then.

The proposed approach for semantic interoperability including the elaboration of the $M \leftrightarrow O$ mappings is currently realized in the *Semantic Culture Guide*[8] project which is conducted by seven partners from industry and academia and funded by the FIT-IT initiative of the Austrian Federal Ministry of Transport, Innovation and Technology.

4 CONCLUSION AND FURTHER WORK

In this paper we have presented an approach for achieving semantic interoperability of business process models. The basic idea of our approach is the combined use of metamodels and ontologies as this is the only means to provide a complete description of the semantics of model elements. While the relationship between model elements and their metamodel is implicitly defined, this is not the case for the relationship between model elements and ontology concepts. Therefore, we have discussed some basic approaches for establishing these links in this paper. The formalization and realization of these

[7] These values have been calculated using WordNet::Similarity which is available at http://marimba.d.umn.edu/cgi-bin/similarity.cgi and described in Pedersen, Patwardhan and Michelizzi (2004).
[8] Please refer to http://wiki.sembase.at/index.php/SCG for more details.

approaches will be a central point of future work during the implementation of the described architecture for interoperability using an existing metamodeling framework. During this implementation we are also working on further theoretical questions concerning the interoperability process. For instance, it has to be clarified to what extent one can make use of ontologies that are semantically strong. Are business rules, for instance, instrumental in further enhancing the interoperability process?

Another interesting question is if the explication of the inherent semantics of business process activities can be used for other applications than model interoperability as well. In this context we see two other use cases: The first one is related with the possible identification of business rules that are relevant for a business processes. Business rules are basically defined as statements about how the business is done, i.e., about guidelines and restrictions with respect to states and processes in an organization (cf. Herbst (1996)). They refer to real business entities and restrictions which are applied on them that altogether can be described by domain or enterprise ontologies. If business processes and their activities are semantically annotated with reference to the same ontologies an automatic detection of relevant business rules can be performed which is enabling a check whether the modeled business process is applying to these defined rules or not. We will also examine how this additional information can be used in the interoperability process.

On the other hand the inherent semantics of activities of business processes can be used in the area of service oriented systems. The additional information that is provided by the linkage to ontology concepts could be combined with approaches for semantic service description and discovery which would enable the correct automatic selection of software services that can support business process steps. Related work concerning this scenario can be found in Karagiannis (2006), for instance.

References

Atkinson, C. and Kühne, T. (2002). Rearchitecting the UML Infrastructure. ACM Trans. Model. Comput. Simul., 12 (4), 290-321.

Atkinson, C. and Kühne, T. (2003). Model-Driven Development: A Metamodeling Foundation. IEEE Software, 20 (5), 36-41.

Berners-Lee, T., Hendler, J. and Lassila, O. (2001). The Semantic Web. Scientific American, 284 (5), 34-43.

Brinkkemper, S., Saeki, M. and Harmsen, F. (1999). Meta-Modelling Based Assembly Techniques for Situational Method Engineering. Information Systems, 24 (3), 209-228.

Bézivin, J. (2005). On the Unification Power of Models. Software and System Modeling, 4 (2), 171-188.

Doan, A., Noy, N. F. and Halevy, A. Y. (2004). Introduction to the Special Issue on Semantic Integration. SIGMOD Record, 33 (4), 11-13.

Ehrig, M., Haase, P., Hefke, M. and Stojanovic, N. (2005). Similarity for Ontologies - A Comprehensive Framework. Proceedings of the 13th European Conference on Information Systems, Information Systems in a Rapidly Changing Economy, ECIS 2005.

Favre, J. - M. (2005). Foundations of Meta-Pyramids: Languages vs. Metamodels - Episode II: Story of Thotus the Baboon. In J. Bézivin and R. Heckel (eds.): Language Engineering for Model-Driven Software Development, Internationales Begegnungs- und Forschungszentrum fuer Informatik (IBFI), Dagstuhl, Germany.

Green, P. and Rosemann, M. (2005). Business Systems Analysis with Ontologies. Idea Publ. Group, Hershey PA.

Gruber, T. R. (1993). A Translation Approach to Portable Ontology Specifications. Knowledge Acquisition, 5 (2), 199-220.

Guarino, N. (1998). Formal Ontology and Information Systems. Proceedings of FOIS'98, p. 3-15, IOS Press, Amsterdam.

Haller, A., Cimpian, E., Mocan, A., Oren, E. and Bussler, C. (2005). WSMX - A Semantic Service-Oriented Architecture. In IEEE Computer Society (eds.): IEEE International Conference on Web Services (ICWS 2005), p. 321-328.

Harel, D. and Rumpe, B. (2000). Modeling Languages: Syntax, Semantics and All That Stuff - Part I: The Basic Stuff. (Working Paper), available at: http://www4.in.tum.de/publ/papers/HR00.pdf, access: 2006-05-26.

He, Y., He, K. and Wang, C. (2005). Research on Semantic Web Service-Oriented MMFI for Complex Information Registration. IEEE International Workshop Service-Oriented System Engineering, SOSE 2005, p. 229-234.

Herbst, H. (1996). Business Rules in Systems Analysis: a Meta-Model and Repository System. Information Systems, 21 (2), 147-166.

IBM (2006). Line of Visibility Enterprise Modeling (LOVEM). Available at: http://www-935.ibm.com/services/us/index.wss/offering/bcs/a1006593, access: 2006-11-24.

Jamadhvaja, M. and Senivongse, T. (2005). An Integration of Data Sources with UML Class Models Based on Ontological Analysis. In A. Hahn, S. Abels and L. Haak (eds.): Proceedings of the First International ACM Workshop on Interoperability of Heterogeneous Information Systems (IHIS'05), CIKM Conference, p. 1-8.

Kalfoglou, Y. and Schorlemmer, M. (2005). Ontology Mapping: The State of the Art. In Y. Kalfoglou, M. Schorlemmer, A. Sheth, S. Staab and M. Uschold (eds.): Semantic Interoperability and Integration, Internationales Begegnungs- und Forschungszentrum fuer Informatik (IBFI), Dagstuhl, Germany.
Kappel, G., Kapsammer, E., Kargl, H., Kramler, G., Reiter, T., Retschitzegger, W., Schwinger, W. and Wimmer, M. (2006). On Models and Ontologies - A Layered Approach for Model-based Tool Integration. In H. C. Mayr and R. Breu (eds.): Modellierung 2006, p. 11-27.
Karagiannis, D. (2006). What Can Web Services Learn from Business Process Modeling? In Y. Yan and L. Zhang (eds.): Proceedings of the 2006 International Workshop on Service Oriented Techniques (SOT06); Eight International Conference on Electronic Commerce (ICEC 2006), p. 1-7.
Karagiannis, D. and Höfferer, P. (2006). Metamodels in Action: An Overview. In J. Filipe, B. Shishkov and M. Helfert (eds.): ICSOFT 2006 - First International Conference on Software and Data Technologies, p. IS-27 - IS-36, Insticc Press, Setúbal.
Karagiannis, D., Junginger, S. and Strobl, R. (1996). Introduction to Business Process Management Systems Concepts. In B. Scholz-Reiter and E. Stickel (eds.): vol. Business Process Modelling, p. 81-106Springer, Berlin Heidelberg.
Kayed, A. and Colomb, R. M. (2005). Using BWW model to evaluate building ontologies in CGs formalism. Information Systems, 30 (5), 379-398.
Keller, G., Nüttgens, M. and Scheer, A. - W. (1992). Semantische Prozeßmodellierung auf der Grundlage "Ereignisgesteuerter Prozeßketten (EPK)". (Working Paper), available at: http://www.iwi.uni-sb.de/nuettgens/Veroef/Artikel/heft089/heft089.pdf, access: 2006-03-10.
Klint, P., Lämmel, R. and Verhoef, C. (2005). Toward an engineering discipline for grammarware. ACM Trans. Softw. Eng. Methodol., 14 (3), 331-380.
Knowledge Based Systems Inc. (1995). Information Integration for Concurrent Engineering (IICE) - IDEF 3 Process Description Capture Method Report. Available at: http://www.idef.com/pdf/Idef3_fn.pdf, access: 2006-11-24.
Kramler, G., Kappel, G., Reiter, T., Kapsammer, E., Retschitzegger, W. and Schwinger, W. (2006). Towards a semantic infrastructure supporting model-based tool integration. GaMMa '06: Proceedings of the 2006 international workshop on Global integrated model management, p. 43-46, ACM Press, New York, NY, USA.
Kühn, H. (2004). Methodenintegration im Business Engineering. PhD Thesis, Universität Wien.
Kühn, H. and Murzek, M. (2006). Interoperability Issues in Metamodelling Platforms. In D. Konstantas, J. - P. Bourriéres, M. Léonrad and N. Boudjlida (eds.): 1st International Conference on Interoperability of Enterprise Software and Applications (I-ESA), Geneva, Switzerland, p. 216-226, Springer, Berlin Heidelberg.
McGuinness, D. L. (2003). Ontologies Come of Age. In D. Fensel, J. Hendler, H. Lieberman and W. Wahlster (eds.): Spinning the Semantic Web - Bringing the World Wide Web to its Full Potential, p. 171-194MIT Press, Cambridge, MA.
Noy, N. F. (2004). Semantic Integration: A Survey Of Ontology-Based Approaches. SIGMOD Record, 33 (4), 65-70.
Object Management Group (2003). MDA Guide Version 1.0.1. Available at: http://www.omg.org/docs/omg/03-06-01.pdf.
Object Management Group (2006a). Business Process Modeling Notation Specification. Available at: http://www.omg.org/docs/dtc/06-02-01.pdf, access: 2006-10-18.
Object Management Group (2006b). Ontology Definition Metamodel (Fifth Revised Submission). Available at: http://www.omg.org/docs/ad/06-01-01.pdf, access: 2006-04-27.
Obrst, L. (2003). Ontologies for Semantically Interoperable Systems. CIKM '03: Proceedings of the Twelfth International Conference on Information and Knowledge Management, p. 366-369, ACM Press, New York, NY, USA.
Pan, J. Z. and Horrocks, I. (2001). Metamodeling Architecture of Web Ontology Languages. In I. F. Cruz, S. Decker, J. Euzenat and D. L. McGuinness (eds.): The Emerging Semantic Web; First Semantic web working symposium, Stanford University, California, USA, July 30 - August 1, 2001, IOS Press, Amsterdam.
Pedersen, T., Patwardhan, S. and Michelizzi, J. (2004). WordNet: : Similarity - Measuring the Relatedness of Concepts. In D. L. McGuinnes and G. Ferguson (eds.): Proceedings of the Nineteenth National Conference on Artificial Intelligence, Sixteenth Conference on Innovative Applications of Artificial Intelligence, p. 1024-1025.
Roser, S. and Bauer, B. (2005). Ontology-Based Model Transformation. In J. - M. Bruel (eds.): MoDELS Satellite Events, p. 355-356, Springer, Berlin Heidelberg.
Smith, B. and Weltym, C. (2001). FOIS introduction: Ontology-towards a new synthesis. FOIS '01: Proceedings of the international conference on Formal Ontology in Information Systems, p.3-9, ACM Press, New York, NY, USA.
Strahringer, S. (1996). Metamodellierung als Instrument des Methodenvergleichs: eine Evaluierung am Beispiel objektorientierter Analysemethoden. Shaker, Aachen.
Studer, R., Decker, S., Fensel, D. and Staab, S. (2000). Situation and Perspectives of Knowledge Engineering. In J. Cuena, Y. Demazeau, A. Garcia Serrano and J. Treur (eds.): Knowledge Engineering and Agent Technology, p. 237-252IOS Press, Amsterdam.
Terrasse, M. - N., Savonnet, M., Leclercq, E., Grison, T. and Becker, G. (2006). Do we need metamodels AND ontologies for engineering platforms? GaMMa '06: Proceedings of the 2006 international workshop on Global integrated model management, p. 21-28, ACM Press, New York, NY, USA.
Uschold, M. and Gruninger, M. (2004). Ontologies and Semantics for Seamless Connectivity. SIGMOD Record, 33 (4), 58-64.
Zaniolo, C. and Melkanoff, M. A. (1982). A Formal Approach to the Definition and the Design of Conceptual Schemata for Database Systems. ACM Trans. Database Syst., 7 (1), 24-59.