Linking Heterogeneous Conceptual Models through a Unifying Modeling Concepts Ontology

Janina Fengel
University of Applied Sciences Darmstadt
Hohe Strasse
D – 64807 Dieburg
+49 6151 16 9458
janina.fengel@h-da.de

Michael Rebstock
University of Applied Sciences Darmstadt
Haardtring 100
D – 64295 Darmstadt
+49 6151 16 8392
michael.rebstock@h-da.de

ABSTRACT
In this discussion paper we report on our ongoing work in applying Semantic-Web technologies for supporting business integration. Our method foresees the reengineering of conceptual models into ontologies for performing domain-semantics oriented matching. For linking models in differing modeling languages as well as different model types, we have developed a bridge ontology. The first results show the feasibility of our approach.

Categories and Subject Descriptors
H.4.0 [**Information Systems**]: Information Systems Applications – *General.*

General Terms
Management, Standardization, Languages.

Keywords
Conceptual modeling, semantic heterogeneity, domain semantics, modeling languages, bridge ontology

1. INTRODUCTION
The need for integrating conceptual models arises at the time of process optimization, business (re-)engineering or generally in business integration. Typical situations are reorganizations or mergers, leading to process and application integration challenges. Upon integrating the conceptual models describing the business operations and the underlying IT-support, heterogeneously used natural language for labeling model element labels often hinders meaningful comparison. Furthermore, the usage of different models in differing modeling languages usually prevents automated support in aligning, linking or merging models. Nevertheless, models to be integrated need to be compared regarding the intended meaning of their elements and their structure, whereas structural analysis cannot be performed until successful alignment of the domain language [13]. Thereby, especially naming conflicts hinder model integration [1; 14]. In practice, often differing unrelated non-aligned legacy models exist. There are efforts in matching models concentrating on the aspect of model language semantics based on migration or transformation from one modeling language into another [12; 7; 10], matching models via their meta models [9] or concentrating on managing models of the same kind [11]. Thereby, the aspect of heterogeneously used domain language is not addressed, instead the model element labels are transferred and retained unchanged for further use. Extending process models with semantic annotations for easing consistent modeling and business-IT alignment has been suggested, thus turning models into model instances [16; 2]. For assigning element labels, the use of a separately developed domain ontology has been proposed, similar as in the suggestions for semantic business process management, which rely on such a pre-defined business terminology [8; 4; 15]. It can assist in the creation of new models and provide the basis for unambiguous element labeling as well as serve for mediating the matching of existing models. However, the creation of a common domain ontology or business terminology to be used as a standard is usually time-consuming and cost-intensive. Furthermore, comparing a model to the set standard is still labour-intensive work.

For easing this workload, we propose to convert existing process, data and organizational models into ontologies and provide automated support for relating them by means of ontology matching techniques. In this, our approach may serve as a complement to the existing works in process matching as outlined, as it offers a means to semantically integrate models of different kind regarding the domain and modeling language together. Additionally, through the semantics-oriented reuse of the domain knowledge contained in models, over time the collection of linked models may be taken as a skeleton semantic domain or, more specifically, enterprise or business ontology. In the following we continue with describing our method of converting models into ontologies, followed by presenting the bridge ontology specifically designed for enabling semantic integration. We conclude with showing the method's application using a small example, closing with a brief discussion and outlook onto our further work.

2. ONTOLOGIZING MODELS
By ontologizing models the business concepts' semantics are made machine-accessible, independent of the modeling languages used. The creation of a model requires knowledge of the domain language for naming the business concepts to be described as well as the modeling language for describing their relations and sorting them. Reversing this process allows us to decompose a model into

SBPM 2010, May 31, 2010, Heraklion, Crete, Greece.

the domain semantics separately from the modeling language semantics [5]. A model and its elements are split into two separate ontologies in OWL DL, which together describe the model with its model type and name and the model elements with their model element types and labels. In the model conversion, all labels are presently taken without further processing, so that not only terms, but complete expressions are transferred. Often, domain knowledge in the field of business processing lies in the combination of objects and the execution of activities, which is preserved this way. Basically, the suggested decomposition method abstracts from the statement a model intends to do and leaves the model as is for further active use. Figure 1 shows the meta model of a thus decomposed conceptual model.

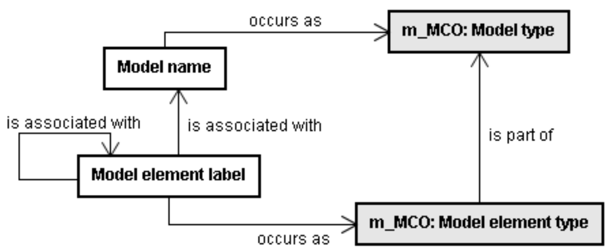


Figure 1. Meta model of a decomposed model

Thereby, the domain facts expressed in the natural domain language are separated from the type of element they are connected with. This type information resembles attaching provenance information. For doing so, the idea of indexing the domain facts in a manner similar to indexing in librarianship in form of Topic Maps has been adopted. The model ontology on the left side captures the domain knowledge in natural language as `owl:Classes` and the relations between them as properties with restrictions as needed. This model ontology links to the model type ontology. For each modeling language a specific Modeling Concept Ontology (MCO) has been developed, containing parts of its meta model. The domain knowledge expressed in the logical relations between model elements as the means for setting the specific models' element order is preserved together with the domain facts in the model ontology, not in the MCO. This conversion returns the element labels representing business concepts as classes, thereby allowing a later ontology extension with the concepts' instances. Thus, the principle of conceptual modeling in business is carried forward.

3. THE UMCO AS A BRIDGE

In principle, for any type of model an MCO can be developed. In order to be able to link the ontologies resulting from the conversion as described, the MCOs enable references between models of the same type. For further enabling also the referencing of models of the same kind, but different type as well as also models of different kind, we use the Unifying Modeling Concepts Ontology (UMCO), which we have developed specifically for this purpose. It provides a unifying model concept for each type of modeling concept with a similar intention. In this, it represents all relevant element types for our approach, usable like a meta-meta model. The UMCO and its MCOs serve for linking modeling concepts without predefining relations beyond part-whole-definitions, as this is done in the model ontologies.

Input for the development of the UMCO has been drawn from existing enterprise modeling languages and ontologies. Business process modeling languages provide the means for describing sequences of activities. They offer the idea of activities, either being called activities, tasks, functions or actions, which start and end with an event and are linked by flows. For the description of the behavioral aspect of processes, the flows can be tied to logical connectors for making decisions and showing alternative flow paths [12]. In detail the semantics of process modeling languages are not equivalent, so that models cannot be translated directly without loss of information [10]. However, the fundamental intensions of the concepts are comparable. The same observation can be made for models describing static business information. Conceptual data models can be represented as entity-relationship models, UML class models or directly as OWL-ontologies. Thereby, the entities of an ERM or classes of a class model correspond to the classes of an ontology, while the attributes and relationships correspond to the relations or properties in most ontology languages [17]. Furthermore, UML class models can be used for ontology modeling [6; 3]. For our purpose of integrating models, we have defined general modeling concepts and declared them equivalent to corresponding concepts in the various modeling languages. In this, the UMCO further extends the meta model shown in Figure 2.

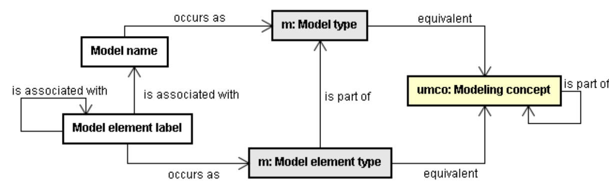


Figure 2. Meta model of a decomposed model with relations to the UMCO

The UMCO is extensible as needed for including further MCO-ontologies. Developing MCOs for process, data and organization models, or any other models, allows connecting knowledge not only directly for model ontologies originating from the same language space of the modeling languages, but additionally, through the unifying modeling concepts, also knowledge from model ontologies of different modeling languages. For example, an EPC is defined compliant with an UML activity model and a BPMN model, and these concepts are set to be equivalent to the UMCO concept called `UMCO:Process`; `EPC_MCO:Function` in an EPC, `UML_AM_MCO:`Action in an UML activity model and `BPMN:Task` in a BPMN model are set to be equivalent to the concept called `UMCO:Activity`. Accordingly, all modeling concepts found in the various languages can be unified and related, e.g. linking resources in process models such as documents or participants to the data models detailing them.

4. LINKING MODELS

Usually, matching models is a major task. To ease this workload, we partially automate this step by reengineering the conceptual domain knowledge contained in existing models as shown and relate the resulting model ontologies semantically for establishing semantic correspondence between the model ontologies' elements. As an implementation for semantic model integration in the described manner, we develop the MODI (Model Integration) Framework as an application of our method. Our framework is realized in Java and can be accessed by a web service interface. It consists of a core component, to which tools for mapping and storage can be variably connected by adapters. Figure 3 shows the architecture.

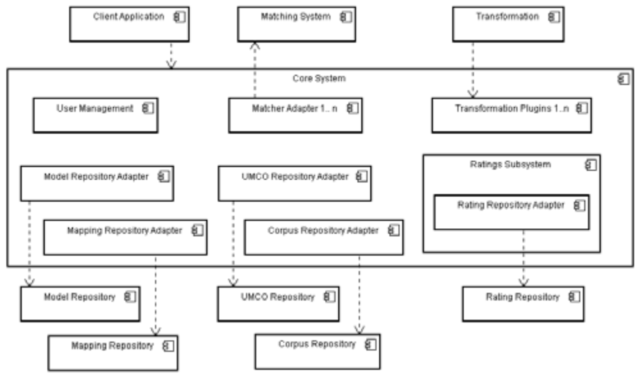


Figure 3. MODI Framework Architecture

The focus of our work is on the realization of matching and establishing domain-semantics based mappings between models. Having performed the model conversions as described above, ontology matching can be performed without merging any of the input model ontologies. As the model ontologies obtained by converting process models do not contain hierarchical or mereologic relations, only element-based techniques return meaningful mappings, best by tokenization and name matching. For matching converted ERM and class model ontologies, also structure-based techniques can be used, as here in most cases subsumption and aggregation relations return exploitable ontology structures. Since the domain facts are not transferred as instances of their individual model type, it is prevented that matchings return mappings between model element types. These links are provided without creating workload for the matchers through the introduction of the MCOs and the UMCO. Furthermore, avoiding such an undesired hierarchical structure focuses the matching efforts onto the domain language independently of the original modeling language used. Figure 4 shows excerpts from two converted business process models from the travel domain as an example. Each model depicts the booking of services. The source model “Travel Reservation” is an EPC, while the target model “Travel Booking” is a UML Activity Model. They both depict the process of booking travel services, however, the domain language differs.

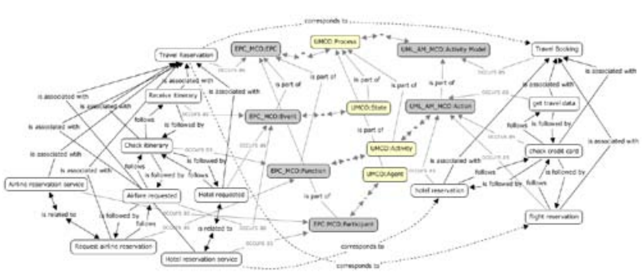
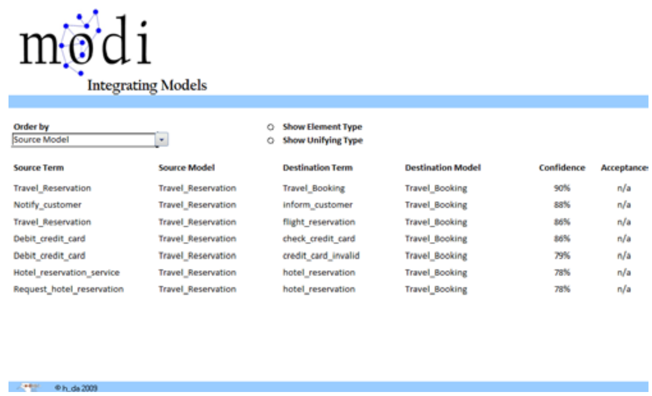


Figure 4. Excerpts from two linked business process models

The matching works could be performed successively as needed. Results became available from the beginning, especially after having included lexical background information from WordNet. In our framework, all mappings found as a result of matchings are stored in a repository as semantic correspondences. Thus, an initial base is being established by means of automated tools. At the time of using the resulting mapping ontology, the automatically derived information is evaluated by its users. Even though the need for manual work is reduced, the mappings found are not always perfect, but may be ambiguous or incorrect [18]. Therefore, our system facilitates user participation by enabling adding, editing and feedback provision for growth and improvement over time. The evolving repository can be queried for semantic correspondences. Thereby, a user may request references for a specific term or directly compare two ontologies. Figure 5 shows the prototypical screen of the results for a comparison of the two example business process models.



Source Term	Source Model	Destination Term	Destination Model	Confidence	Acceptance
Travel_Reservation	Travel_Reservation	Travel_Booking	Travel_Booking	90%	n/a
Notify_customer	Travel_Reservation	inform_customer	Travel_Booking	88%	n/a
Travel_Reservation	Travel_Reservation	flight_reservation	Travel_Booking	86%	n/a
Debit_credit_card	Travel_Reservation	check_credit_card	Travel_Booking	86%	n/a
Debit_credit_card	Travel_Reservation	credit_card_invalid	Travel_Booking	79%	n/a
Hotel_reservation_service	Travel_Reservation	hotel_reservation	Travel_Booking	78%	n/a
Request_hotel_reservation	Travel_Reservation	hotel_reservation	Travel_Booking	78%	n/a

Figure 5. Prototypical list of suggested semantic correspondences

With an increasing number of models included, first tendencies towards commonly used terms can become obvious. An initial terminological domain ontology emerges, consisting of the various independent model ontologies, which are linked through the mappings stored. This emerging ontology can be used at the time of creating new models searching for a suitable element label as well as for explaining the intended meaning in existing models that need to be compared and related semantically.

Combining the model ontologies with our modeling concepts ontologies allows for searching for specific model types., e.g., searching for all EPCs available, as well as for models of all types of a certain kind, e.g., such as process models either being EPC or UML Activity Models, through utilizing the corresponding unifying concept, here `UMCO:Process`, for detailing the query. Alternatively, searches for `UMCO:Activity` return all business operation steps. With the method described, not only models of the same kind can be matched and related. Instead, linking different models is possible as well with the help of the various MCOs und the UMCO.

5. DISCUSSION AND CONCLUSION

Here we proposed an approach based on applying ontology engineering techniques for achieving semantic integration of conceptual models in the business domain. A method for reusing existing conceptual models and relating the business knowledge contained without huge manual efforts is shown. We have created a way for reengineering such non-ontological resources for meaningful relating and linking with the help of supporting ontologies especially created for this purpose. The related models can be analyzed and compared regarding the intended meaning of their elements. The automatically produced results provide a basic lightweight domain ontology without initial manual preparation and creation efforts. By including user input, a possibility for overcoming the shortcomings of automated knowledge computing is presented, as human support is included for assessing and improving the quality of the mappings found by way of automated matching. The resulting mapping collection provides support for the clarification of uncertainties and allows for semantically integrating models of any type. Our system works as a mediating medium and helps providing the grounds for concentrating on the actual questions of process integration and activity sequencing.

The system presented here has been implemented as a prototypical solution for the method developed and is being evaluated. Its application has proven the conceptual strength and practical relevance. Still, a number of aspects remain to be researched. Since the system is based on coupling existing tools for ontology matching and storing via adapters, evaluation concerning their efficiency and performance is of interest. The research concerning the comparison and combination of mapping tools has to be concluded. Furthermore, as also is the case with social software, our system needs a critical mass of users in order to be useful. Hence, it needs to be proven that the method and with this also our framework offers benefits. Overall, with our approach of semantic model integration we hope to have shown how the usage of Semantic-Web technologies may support business modeling.

6. ACKNOWLEDGMENTS

Our work is supported by the German Federal Ministry of Education and Research (BMBF) under grant no. 1728X07 MODI.

7. REFERENCES

[1] Becker, J., Rosemann, M., and Schütte, R. 1996. Prozeßintegration zwischen Industrie- und Handelsunter-nehmen - eine inhaltlich-funktionale und methodische Analyse. Wirtschaftsinformatik 39(3), 309–316.

[2] Belecheanu, R., Cabral, L., Domingue, J., Gaaloul, W., Hepp, M., Filipowska, A., Kaczmarek, M. et al. 2007. Business Process Ontology Framework. Deliverable 1.1. SUPER Project.

[3] Brockmans, S., Volz, R., Eberhart, A., and Löffler, P. 2004. Visual Modeling of OWL DL Ontologies Using UML. In Proceedings of the Third International Semantic Web Conference ISWC 2004 (Hiroshima, Japan, Nov. 2004). Springer, Berlin, 198–213.

[4] Brockmans, S., Ehrig, M., Koschmider, A., Oberweis, A., and Studer, R., 2006. Semantic Alignment of Business Processes. In Proceedings of the Eighth International Conference on Enterprise Information Systems ICEIS 2006 (Paphos, Cyprus, May 2006). INSTICC, Setúbal, 197–203.

[5] Fengel, J. and Rebstock, M., 2009. Model-Based Domain Ontology Engineering. In Proceedings of the 4th International Workshop on Semantic Business Process Management SBPM2009 (Heraklion, Crete, June 2009). ACM Press, New York.

[6] Gašević, D., Djurić, D., Devedžić, V., and Damjanović, V., 2004. Converting UML to OWL ontologies. In Proceedings of the thirteenth International World Wide Web Conference. ACM Press, New York, 488–89.

[7] Gehlert, A., 2007. Migration fachkonzeptueller Modelle. Logos-Verl., Berlin.

[8] Hepp, M., Leymann, F., Domingue, J., Wahler, A., and Fensel, D., 2005. Semantic Business Process Management: A Vision Towards Using Semantic Web Services for Business Process Management. In Proceedings of the IEEE International Conference on e-Business Engineering ICEBE 2005 (Beijing, China, Oct. 2005). IEEE Computer Society, Los Alamitos, 535–40.

[9] Kappel, G., Kramler, G., Kapsammer, E., Reiter, T., Retschitzegger, W., and Schwinger, W., 2005. ModelCVS - A Semantic Infrastructure for Model-based Tool Integration. Technical Report. University of Vienna.

[10] Kensche, D., Quix, C., Chatti, M. A., and Jarke, M., 2007. GeRoMe: A Generic Role Based Metamodel for Model Management. Journal on Data Semantics VIII, LNCS 4380, Springer, Berlin, 82–117.

[11] Melnik, S. Rahm, E., and Bernstein, P. A., 2003. Rondo: A Programming Platform for Generic Model Management. In Proceedings of the ACM SIGMOD International Conference on Management of Data (San Diego, California, June, 2003). ACM Press, New York, 193–204.

[12] Murzek, M. and Kramler, G., 2007. The Model Morphing Approach – Horizontal Transformations between Business Process Models. In Proceedings of the 6th International Conference on Perspectives in Business Information Research - BIR'2007. Vol. D-2007-13. Tampere, Finland, Department of Computer Sciences, 88–103.

[13] Simon, C. and Mendling, J., 2007. Integration of Conceptual Process Models by the Example of Event-driven Process Chains. In Proceedings of Wirtschaftsinformatik WI 2007 (Karlsruhe, Germany, March 2007). Vol. 1. Univ.-Verl., Karlsruhe, 677–94.

[14] Thomas, O. and Fellmann, M. 2006. Semantische Integration von Ontologien und Ereignisgesteuerten Prozessketten. In Proceedings of EPK 2006 - Geschäftsprozessmanagement mit Ereignisgesteuerten Prozessketten (Vienna, Austria, Nov. 2006). Vol. 224. CEUR-WS.org, 7–23.

[15] Thomas, O. and Fellmann, M. 2007. Semantic Business Process Management: Ontology-Based Process Modeling Using Event-Driven Process Chains. IBIS Interoperability in Business Information Systems 2(1), 29–44.

[16] Thomas, O. and Fellmann, M., 2009. Semantische Prozess-modellierung – Konzeption und informationstechnische Unterstützung einer ontologiebasierten Repräsentation von Geschäftsprozessen. Wirtschaftsinformatik 51(6), 506–518.

[17] Uschold, M. and Gruninger, M., 2004. Ontologies and Semantics for Seamless Connectivity. ACM SIGMOD Record 33(4), 58–64.

[18] Zhdanova, A J. de Bruijn, K. Zimmermann, and F. Scharffe, 2004. Ontology Alignment Solution v2.0. Deliverable D1.4 V2.0, Esperonto project.