FROM CONCEPTUALIZATION TO MODEL GENERATION: THE ROLES OF ONTOLOGIES IN PROCESS MODELING

Aidong Yang, Jan Morbach, Wolfgang Marquardt
Lehrstuhl für Prozeßtechnik, RWTH
Turmstr. 46, D-52064 Aachen, Germany

Abstract

The expanding scope of systems tackled by process engineering and the increasing interest in different scales of systems require process modeling tools to be generic and extensible. These requirements are addressed in this work by employing ontologies in two different modeling phases. Firstly, a domain ontology providing a conceptualization of process systems is used by a conceptual modeling tool to generate conceptual process models. They are composed of instances of concepts and their relations as defined in the domain ontology. Secondly, a mathematical model generation engine is implemented using a general systems meta ontology. Employing the knowledge embodied in the meta ontology, this engine takes a conceptual process model as input and composes a mathematical model by means of selecting and customizing model building blocks stored in a library. In addition to the above roles of ontologies in developing models, it is also briefly discussed how ontology-based modeling can support the exchange of information between tools for modeling and design.

Keywords

Process modeling, Ontologies, Conceptual modeling, Model generation, Process design.

Introduction

To support the development of mathematical models of process systems, a number of efforts have been made on computer-aided process modeling (CAPM) (Marquardt, 1996; see also references in Yang and Marquardt, 2004). Currently, models have been developed for various types of process systems ranging from chemical processing systems to biological systems and from continuous systems to discrete-event or hybrid systems. Further, different scales of a system may be considered to study the corresponding behavior. The continuously expanding scope and the increasing interest on properly modeling different scales of process systems imply that it is impractical to build a CAPM tool once, which is then expected to serve the needs of any modeling context not anticipated during the development of the CAPM tool. Instead, the tool has to be generic and highly extensible. By being generic, there would be no need to change the core functionality of the tool when changes occur in the type or scale of the objects it can handle. By being extensible, the above changes could be tackled with little effort and without limitations imposed by the tool architecture.

In this work, we have explored how to address the above issues by making use of ontologies. An ontology is an explicit specification of a conceptualization, typically involving concepts, their relations, and axioms for clarifying intended semantics (Uschold & Gruninger, 1996). An ontology can be represented either informally using e.g. natural language or formally using a certain computer language. In the latter case, the ontology can be processed automatically by some software tools. Depending on the subject of conceptualization, different types of ontologies exist; among those are domain ontologies, which provide conceptualization of a certain domain (e.g. chemical engineering, biological engineering, etc.), and meta ontologies, which provide a set of highly abstract and generic concepts independent of any concrete domains (cf. van Heijst et al., 1997). In the context of process modeling, Bogusch et al. (2001) discuss the potential role of ontologies in conceptualizing chemical processes. Batres et al. (2002) use ontologies to improve the exchangeability of mathematical models.

In this paper, we discuss the usage of ontologies for the purpose of improving the generality and extensibility of CAPM tools. In doing so, we distinguish between two sequential phases in modeling: *conceptual modeling* and *mathematical model generation and analysis*, as done in a number of recent CAPM efforts (cf. references in Yang and Marquardt, 2004). In conceptual modeling, the modeler characterizes a system with physicochemical concepts. After that, s/he goes on to the second phase to formulate and analyze mathematical equations.

In Section 2, we present an ontology-based approach for conceptual modeling and a realization of this approach. Mathematical model generation using a meta ontology is addressed in Section 3. In Section 4, we briefly discuss the implications of our approach on the interaction of process modeling and design.

Conceptual Modeling Using a Domain Ontology

In general, formulation of a conceptual model involves two aspects: the provision of concepts needed for process modeling and the instantiation of these concepts according to the concrete process systems to be modeled. In an ontology-based modeling approach (Yang and Marquardt, 2004), the role of an ontology is to provide concepts to describe the behavior of process systems. In addition, an ontology can also impose constraints on the semantics of some concepts in order to clarify their intended meaning and/or usage. Such constraints, often called axioms, are especially useful in process modeling to avoid potential modeling mistakes at the "physical level" prior to the derivation of mathematical equations.

Process modeling concepts provided by an ontology have to be instantiated to generate concrete conceptual models. This requires interactions between a human modeler and a conceptual modeling tool (CMT). Figure 1 presents a structure of a CMT, which makes use of general ontology tools developed in the ontological engineering community. A graphical interface provides the access to an ontology querying tool for retrieving those concepts from the domain ontology which are useful for the current modeling activity. It further allows for the modeler to declare instances and their connections. Once a conceptual model is composed, it can be sent to a tool for checking its consistency with respect to the ontology employed.

Based on an existing CAPM system (Hackenberg, 2004), a prototype has been developed according to the above tool structure. The prototype uses OntoCAPE as domain ontology. OntoCAPE is developed in the COGents project (Braunschweig et al, 2004) on the basis of CLiP, a comprehensive conceptual data model for process engineering (Bayer and Marquardt, 2004). OntoCAPE is represented by means of the formal ontology modeling language DAML+OIL (www.daml.org). It consists of a number of modules, covering process modeling as well as model-based activities such as simulation and design. As an example, the UML class diagram in Figure 2 gives an overview on some high level concepts representing the behavioral aspect of the (physicochemical) processing subsystem of a chemical process system.

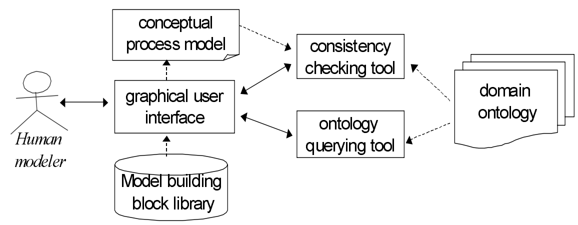

Figure 1. Structure of a conceptual modeling tool.

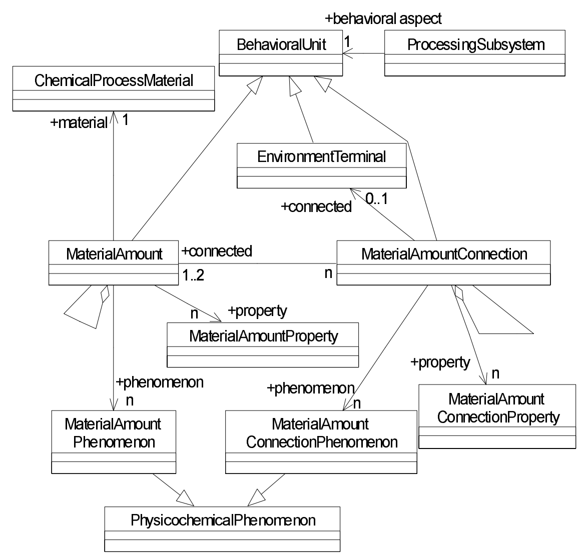

Figure 2. High level OntoCAPE concepts describing processing subsystems.

In another part of OntoCAPE not shown here, a class hierarchy of physicochemical laws and models for calculating individual physicochemical properties is formulated. When creating a conceptual model, certain instances of physicochemical laws or property models, called *model building blocks* (cf. Figure 1), are chosen for modeling phenomena or properties. Note that the phase of conceptual modeling only determines which laws/property models (formulated in certain model building blocks) are adopted; the retrieval (and customization) of those blocks will be handled in the phase of mathematical model generation.

The construction of conceptual models can be further explained using Figure 3. Note that all the elements shown in this figure are for illustrative purposes only. Ignoring the meta ontology part for now, one can see the concepts (e.g. *MaterialAmount*) defined in a *domain ontology* (such as

OntoCAPE, see upper left of Figure 3) are instantiated to represent concrete objects in a *conceptual model* (e.g. *liquid_in_R101*, see lower left of Figure 3). Another example of this kind of instantiation relationship is that between *ReactionRateCoefModel* (in the *domain ontology*) and *Arrhenius_equation* (in the *conceptual model*). As an example of the relation between laws/property models chosen by a conceptual model and the model building blocks, the term *Arrhenius_equation* mentioned above is in fact selected from the (model) *building blocks library* (see upper right of Figure 3).

The prototypical CMT has been developed by using an ontology parser called Jena (www.hpl.hp.com/semweb/) for parsing and querying OntoCAPE, as well as an ontology reasoner called RACER (http://www.sts.tu-harburg.de/~ra.moeller/racer/) for consistency checking. Thus, the main functionality of this CMT is provided by general purpose ontology tools, which simplifies the implementation to a large extent. Further, this CMT is highly generic: its functionality does not depend on the contents of the specific domain ontology used (such as OntoCAPE). The only requirement on the ontology is that it can be abstractly considered as a set of inter-related concepts, regardless of what those concepts and relations actually are. From an extensibility point of view, different types of process systems of interest or different scales of resolution in the lifecycle can be handled, just by adapting or extending OntoCAPE or using some other ontologies.

Meta Ontology Based Automatic Model Generation

Once a *conceptual model* of a certain process system is generated by a CMT, a *mathematical model* can be derived from it. For this purpose a library of model building blocks (already mentioned in the last section) is needed to provide a declarative formulation of concrete physicochemical laws and property models of the domain considered. With both the conceptual model of a certain system and a building block library as inputs, two steps are naturally needed to generate the mathematical model for the target system: (1) retrieval of the building blocks corresponding to the laws and property models stated in the conceptual model, and (2) customization of the selected building blocks from its generic form in order to fit them to the concrete objects described in the conceptual model.

Figure 3 illustrates a prototypical implementation of a model generation engine (MGE) that can perform these two steps. To make this MGE capable of generating models for different types of process systems, we introduce the concept of a *meta ontology*. On top of the domain ontology taken as an input by the CMT, a *meta ontology* provides a set of abstract concepts of general systems, presenting the general relations among a system, its properties, states, events, phenomena, and laws (cf. Figure 3, top left). A *domain ontology* is an instantiation of the meta ontology, defining a concrete type of process system, its parts, and its characteristics (cf. Figure 3, left). In turn, a *conceptual model* is an instantiation of a domain ontology, as already mentioned earlier. Within this hierarchy of conceptualization, the MGE relies on the meta ontology only. Its role in the MGE is to denote how a conceptual model should be "understood", i.e. what are the general types of elements the MGE can expect from a conceptual model and what are the possible types of relations between those elements. This allows the MGE to apply a generic strategy to analyze the conceptual models and then retrieve and customize the building blocks accordingly.

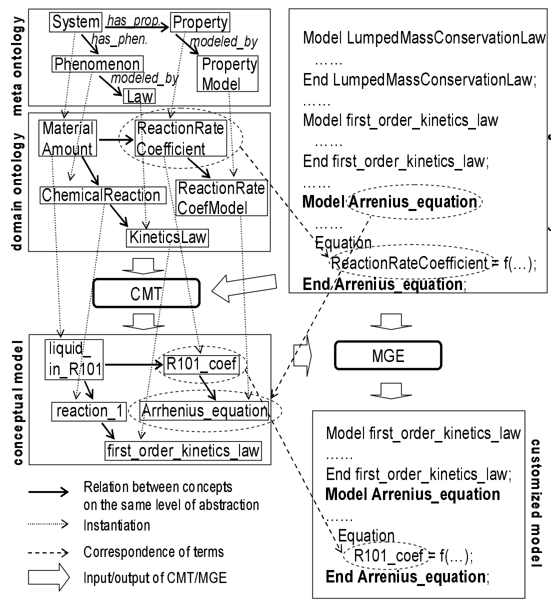

Figure 3. From conceptual modeling to model generation.

Below we illustrate how the MGE takes a conceptual model as input to perform the first step of model generation, i.e. the retrieval of adopted building blocks. According to the concepts in the meta ontology shown in Figure 3, the MGE understands that firstly it should check what "systems" have been declared in the conceptual model. For each detected "system", it checks what "phenomena" and "properties" exist. Then, it checks what "law" or "property model" has been adopted. Consequently, the MGE figures out what (physical item) is modeled by what (building block). It then uses this result to retrieve the selected blocks to accomplish this step.

In the second step, i.e. the customization of the building blocks selected, the MGE utilizes a *building block library* (see Figure 3, top right) formulated in the Modelica language (www.modelica.org). In order to enable automatic customization, the formulation of model building blocks makes use of the concepts in the domain ontology. Particularly, their variables are named corresponding to certain system properties defined in the ontology (e.g. see variable *ReactionRateCoefficient* in model "Arrenius_equation" in Figure 3, middle right). One can see that after the customization, a model variable corresponding to a property defined in the domain ontology (e.g. *ReactionRateCoeffient* in model "Arrhenius_equation") is replaced by an instance of this property class appearing in the conceptual model (e.g. *R101_coef*). Thanks to (a) the correspondence between domain ontology concepts and the variables in the model building blocks, and (b) the relation between the domain ontology and the conceptual model, this automatic replacement is fairly easy. Other more complex treatments, such as replacing a vector equation with specific scalar equations and connecting models of interrelated systems, are also handled in the prototype. As the output of the MGE, a set of customized model building blocks is obtained, which well correspond to the conceptual model.

The implementation of this prototype MGE utilizes a Modelica parser to process model building blocks and the ontology parser Jena to process conceptual models.

Interactions between Process Modeling and Design

Below we will briefly discuss how the use of ontologies for process modeling could improve the interaction between modeling and design. Two important types of tools are used during the conceptual design phase of a chemical plant: process modeling tools for the development and solution of mathematical models, and computer-aided engineering (CAE) systems for the generation, storage, and management of design data. Strong dependencies between mathematical models and design data have been identified (Bayer et al., 2003): design data generated by a CAE system can be used to formulate a skeleton mathematical model in a process modeling tool; in return, new information can be extracted from a simulation with the model to drive modifications and refinement of the design. Therefore, it is desirable to integrate CAE systems and modeling tools to support the bidirectional exchange of information between them.

Such integration requires the definition of mappings between design data and mathematical models. In order to specify these mappings, the information must be modeled explicitly on both the CAE side and the process modeling side. For the latter, conceptual models generated by the CMT provide exactly what is needed: since the conceptual model is built by instantiating a domain ontology, it is available in a well-defined and machine-readable format, which is well suited for specifying the mappings between model and design data. Moreover, if the CAE shares the same conceptualization as the CAPM tool, which is the case when both are supported by a domain ontology like OntoCAPE, the integration can even become easier.

Conclusions

Ontologies can play two major roles in developing process models. A domain ontology of process systems serves as a source of concepts to be used for building a conceptual process model. A conceptual modeling tool can be developed, which is able to take different domain ontologies as inputs for constructing conceptual models. For generating the mathematical model from a conceptual model, a meta ontology can be used to provide the conceptual basis for a model generation engine capable of generating models of different types of systems.

These roles of ontologies bring some new perspectives to the methods and realization of CAPM. Firstly, the multi-level abstraction, especially the separation of a meta ontology from domain ontologies, brings the possibility of adopting a domain-independent strategy for model generation. Secondly, the declarative representation of domain concepts by means of domain ontologies, together with model building blocks formulated accordingly, can make the extension of the scope of a modeling tool simply a matter of resetting its inputs. Finally, the realization of modeling tools can be simplified to a large extent by utilizing generic ontology tools. Beyond the development of models, the use of ontology-based conceptual process modeling provides potential in improving the information sharing and reuse between process modeling and design.

Acknowledgements

This work has been supported by the DFG in the Collaborative Research Center IMPROVE (SFB 476) and by the EC in the COGents project (IST-2001-34431).

References

Batres, R., Aoyama, A., Naka, Y. (2002). A life-cycle approach for model reuse and exchange. *Comput Chem Engng, 26,* 487.

Bayer, B., Marquardt, W. (2004). Towards integrated information models for data and documents. *Comput Chem Engng,* in print.

Bayer, B., von Wedel, L., and Marquardt, W. (2003). An integration of models and design data. In: A. Krawslawski, I. Turunen (Eds.): *"European Symposium on Computer Aided Process Engineering -13"*, Elsevier, 2003, 29.

Bogusch, R., Lohmann, B., Marquardt, W. (2001). Computer-aided process modeling with MODKIT. *Comput Chem Engng, 25,* 963.

Braunschweig, B., Fraga, E., Guessoum, Z., Marquardt, W., Nadjemi, O., Paen, D., Pinol, D., Roux, P., Serra, M., Stalker, I., and Yang, A. (2004). CAPE web services: the COGents way. Accepted by *14th European Symposium on Computer-Aided Process Engineering*.

Hackenberg, J. (2004). *Computer support to theory-based modeling of process systems.* PhD Thesis, Lehrstuhl für Prozesstechnik, RWTH Aachen. In preparation.

Marquardt, W. (1996). Trends in Computer-Aided Process Modeling. *Comput Chem Engng, 20,* 591.

Uschold, M. & Gruninger, M. (1996). Ontologies: principles, methods and applications. *Knowl Engng Rev, 11,* 93.

van Heijst, G., Shreiber, A., and Wielinga, B. J. (1997). Using explicit ontologies in KBS development. *Int. J. Human – Computer Studies, 45,* 183.

Yang, A. and Marquardt, W. (2004). An ontology-based approach to conceptual process modeling. Accepted by *14th European Symposium on Computer-Aided Process Engineering.*