

USING OMG'S SYSML TO SUPPORT SIMULATION

Christiaan J. J. Paredis
Thomas Johnson

Product and Systems Lifecycle Management Center,
Georgia Institute of Technology,
Atlanta, GA 30332. U.S.A.

ABSTRACT

Currently, system engineering problems are solved using a wide range of domain-specific models and corresponding languages. It is unlikely that a single unified modeling language will be able to model in sufficient detail the large number of system aspects addressed by these domain-specific languages. Instead, a model integration framework is needed for managing the various modeling languages used to solve systems engineering problems. The Systems Modeling Language (OMG SysML™) can provide an answer to this need for model integration. Using SysML, a modeler can abstract a domain-specific language to a level that permits its interaction with other system models. In addition, graph transformation approach can be used to accomplishing automated, bidirectional transformation between SysML and the domain specific language. In this paper, a generic approach for defining such graph transformations is presented.

1 INTRODUCTION

Contemporary systems engineering projects are becoming increasingly complex as they are handled by geographically distributed design teams, constrained by the objectives of multiple stakeholders, and inundated by large quantities of design information. Accordingly, problems encountered during the system development process generally have more to do with the organization and management of complexity than with the direct technological concerns that affect individual subsystems and specific physical science areas (Sage 2000). If engineers cannot efficiently manage project complexity, they might overlook important design details and dependencies. Such mistakes can compromise stakeholder objectives and lead to costly design iterations or system failures.

According to the principles of model-based systems engineering (MBSE) (Fisher 1998), engineers can overcome these problems by replacing document-centric design methods with model-based approaches for

representing and investigating their knowledge during system decomposition and definition. Models can be used to represent formally all aspects of a systems engineering problem, including the structure, function, and behavior of a system (Gero 1990). Additionally, experiments can be performed on models to eliminate poor design alternatives and to ensure that a preferred alternative meets the stakeholders' objectives. Models also facilitate collaboration by providing a common, unambiguous protocol for communicating design information.

Although the current focus of MBSE has been on describing the structure, function and behavior of systems, there is an important opportunity to extract from these formal descriptions executable models — i.e., *simulations*. In this paper, we explore a method based on graph transformations to translate — or compile — structural descriptions of a system into corresponding simulation models ready to be executed in a simulation tool.

2 A GRAPH TRANSFORMATION APPROACH

Assume that the dynamic behavior of a system is described in a particular modeling and simulation tool. Since this simulation model is only one aspect of the overall system description, it is important to define formally how the different parameters in the simulation model relate to the parameters used in other system descriptions. Without such formal relationships, it is difficult to maintain consistency among all the system descriptions. In addition, the results of the simulation should also be mapped back to the system requirements and objectives.

To achieve such an integration between the behavioral system model and the rest of the system information, we propose to use the following generic integration approach:

1. Create a meta-model for the specific domain that is being integrated.
2. Create a SysML profile in which the domain specific constructs are mapped to stereotypes, which extend closely resembling, existing SysML modeling elements.

3. Create graph transformations relating the domain specific meta-model and the corresponding SysML profile
4. For each of the transformations that update the domain-specific model, include the corresponding API calls to the domain specific tool.

So far we have demonstrated this approach for integrating the Modelica language (Mattsson, Elmqvist, and Otter 1998) with SysML (Johnson, Paredis, and Burkhart 2008). Modelica is a formal language for describing the energy flow through systems in terms of differential-algebraic equations and discrete events. We are currently in the process of applying this same graph transformation approach for integrating with SysML Matlab Simulink models and discrete event models in eM-Plant.

2.1 The Meta-Models

When capturing the information in a particular domain, it is best to do so formally by first defining a meta-model — a model that defines a language that defines the types of information that are relevant in the domain and the relationships between these types of information. Many domains have already been characterized by such formal meta-models, but in other domains meta-models may be lacking or are encoded implicitly in a domain-specific tool (e.g., a modeling or simulation tool). Even if a meta-model does not exist explicitly it is often not difficult to define one and to relate it through API calls to the data that is implicitly organized in a domain specific tool. A simplified example of such a meta-model for the Modelica language is shown in Figure 1.

To enable a mapping between the domain-specific meta-model and SysML, we establish a corresponding meta-model in SysML. However, rather than modifying the SysML meta-model directly, we use the UML profiling mechanism. Profiles have the advantage that they do not modify the underlying meta-model so that SysML tool support is maintained (Weisemöller and Schürr 2008).

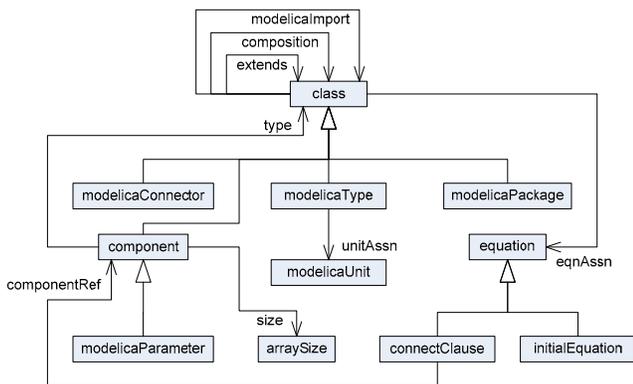


Figure 1: A simplified meta-model for Modelica

2.2 The Graph Transformations

Once the domain-specific meta-models and the corresponding SysML profile have been defined, a mapping or transformation between the two needs to be defined. Many methods exist for completing model transformations between two or more modeling languages or meta-models. Two common transformation approaches are OMG’s Queries/Views/Transformations (QVT) (Object Management Group 2007) and Triple Graph Grammars (TGG) (Schürr 1994). We propose using Triple Graph Grammars.

TGGs are similar to QVT in intent but are declarative by nature. Accordingly, TGGs are particularly useful for completing complex, bidirectional model transformations. In a TGG, the metamodels for the source and target languages are defined as graphs. The mapping between the two languages is then represented as a set of graph transformation rules applied to a third graph: a *correspondence graph*, as is illustrated in Figure 2. Due to the declarative, bidirectional nature of TGGs, one set of graph transformation rules can be used to transform SysML models into the domain-specific language and vice-versa. Although a TGG is used for this transformation, others have shown that QVT is equally expressive and capable (Greenyer 2007).

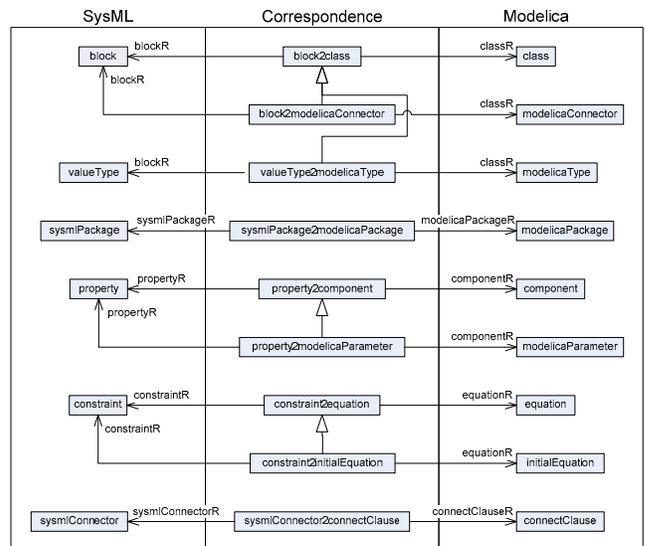


Figure 2: A simplified TGG between Modelica and SysML

3 DISCUSSION AND SUMMARY

We have outlined an approach for integrating domain-specific models into SysML, and have illustrated this approach for an integration between SysML and Modelica — for a more detailed overview refer to (Johnson 2008). Although we have only demonstrated this approach for

Modelica, we believe that it is more generally applicable, and we are currently in the process of illustrating this same approach for integrating Matlab Simulink and eM-Plant into SysML.

Note that when integrating modeling and simulation tools, it may be necessary to distinguish clearly between the models and the simulations. The models describe the behavior while the simulations are experiments performed on the model. It may be necessary to define separate meta-models and model transformations for both the models and the simulations.

ACKNOWLEDGMENTS

The authors would like to acknowledge Lockheed Martin Corporation, and Deere and Company for their support of this research. The ideas in this paper further benefited from constructive criticisms from Roger Burkhart, Sandy Friedenthal, and Russell Peak.

REFERENCES

- Fisher, J. 1998. Model-Based Systems Engineering: A New Paradigm. *INCOSE Insight* 1(3): 3-16.
- Gero, J. S. 1990. Design Prototypes: A Knowledge Representation Schema for Design. *AI Magazine* 11(4): 26-36.
- Greenyer, J., and E. Kindler. 2007. Reconciling TGGs with QVT. *Model Driven Engineering Languages and Systems, MoDELS 2007*. Berlin/Heidelberg, Springer.
- Johnson, T. A. 2008. *Integrating Models and Simulations of Continuous Dynamic System Behavior into SysML*. M.S. Thesis. G.W. Woodruff School of Mechanical Engineering, Georgia Institute of Technology. Atlanta, GA.
- Johnson, T. A., C. J. J. Paredis, and R. M. Burkhart. 2008. Integrating Models and Simulations of Continuous Dynamics into SysML. In *Proceedings of the 6th International Modelica Conference*, 135-145. Bielefeld, Germany, Modelica Association.
- Mattsson, S. E., H. Elmqvist, and M. Otter. 1998. Physical System Modeling with Modelica. *Control Engineering Practice* 6: 501-510.
- Object Management Group. 2007. Meta Object Facility (MOF) 2.0 Query/View/Transformation Specification.
- Sage, A. P., and J. E. Armstrong Jr. 2000. *Introduction to Systems Engineering*. New York, NY, John Wiley & Sons, Inc.
- Schürr, A. 1994. Specification of Graph Translators with Triple Graph Grammars. *WG'94 Workshop on Graph-Theoretic Concepts in Computer Science*.
- Weisemöller, I., and A. Schürr 2008. A Comparison of Standard Compliant Ways to Define Domain Specific Languages. In *Models in Software Engineering: 47-58*, Berlin/Heidelberg: Springer.

AUTHOR BIOGRAPHIES

CHRIS PAREDIS Ph.D., is an Associate Professor in the G.W. Woodruff School of Mechanical Engineering at Georgia Tech. He received his M.S. degree in Mechanical Engineering from the Catholic University of Leuven (Belgium) in 1988, and his M.S. and Ph.D. in Electrical and Computer Engineering from Carnegie Mellon University in 1990 and 1996, respectively. From 1996 to 2002, he was a Research Scientist at the Institute for Complex Engineered Systems at Carnegie Mellon University. Dr. Paredis has a broad, multidisciplinary background. In his research, he combines aspects of information technology, simulation, and systems theory to support the design of mechatronic systems, focusing in particular on decision making under uncertainty in conceptual design. In these areas, he has published more than 80 refereed journal articles and conference papers. Dr. Paredis received the 2007 CETL/BP Junior Faculty Teaching Excellence Award and the 2007 SAE Ralph R. Teeter Educational Award. In 2007-2008, he was the Chair of the ASME Computers and Information in Engineering (CIE) Division. His email address is [<chris.paredis@me.gatech.edu>](mailto:chris.paredis@me.gatech.edu).

THOMAS JOHNSON is a graduate research assistant at the Systems Realization Laboratory at Georgia Tech, where he is working towards his Master's degree in Mechanical Engineering. His research is focused on formal modeling for model-based systems engineering, combining SysML and Modelica. His email address is [<gtg378k@prism.gatech.edu>](mailto:gtg378k@prism.gatech.edu).