SYSTEM ENTITY STRUCTURE AND MODEL BASE FRAMEWORK IN MODEL BASED ENGINEERING OF SIMULATIONS FOR TECHNICAL SYSTEMS

Umut Durak
German Aerospace Center (DLR)
Institute of Flight Systems
Lilienthalplatz 7 38108
Braunschweig, Germany
umut.durak@dlr.de

Thorsten Pawletta
Wismar University of Applied Sciences
FIW / MVU, FG CEA
PF 1210 23952
Wismar, Germany
thorsten.pawletta@hs-wismar.de

Halit Oguztuzun
Middle East Technical University
Department of Computer Engineering
Dumlupınar Bulvarı No:1 06800 Çankaya
Ankara, Turkey
oguztuzn@ceng.metu.edu.tr

Bernard P. Zeigler
RTSync Corp.
and
Arizona Center for Integrated M&S,
Tucson, AZ 85621
zeigler@rtsync.com

ABSTRACT

Model-based engineering is defined as the pragmatic utilization of model-based practices, namely, modeling, metamodeling and model transformations in various steps of engineering. In the last decade, the simulation of technical systems has leveraged graphical modeling and model-to-text transformations, but metamodeling and model transformation practices have not become widely accessible. Thereby, the benefits of employed model-driven approaches have been limited. System Entity Structures are directed labelled graphs that were proposed in the 1980s for specifying a family of system configurations for modular and hierarchical systems. The System Entity Structure and Model Base (SES/MB) framework combines the SES ontology with the classical workflow of modeling for an interactive or automatic generation of executable simulation models. After reviewing comparable approaches in software engineering, this paper discusses the application of SES/MB framework for metamodeling and model transformations for model-based engineering of simulations of technical systems.

Keywords: system entity structures, model-based engineering, metamodeling, model transformations.

1 INTRODUCTION

The term technical systems refers to all man-made machines (Hubka and Elder 1998). The evolution of technical systems brought us to real-time complex designs that are composed of distributed networked heterogeneous physical devices and computational components (Xiao and Fan 2011). This integration of computation with the physical processes came along with various challenges (Lee 2008). Their design became a demanding task which employs mathematical modeling for physical systems and formal modeling for computations, simulating these heterogeneous systems, software synthesis, verification, validation and testing (Jensen et al. 2011). Using models as core assets for designing, analyzing, verifying and validating these complex systems is known as model-based engineering.

SpringSim-Mod4Sim 2017, April 23-26, Virginia Beach, VA, USA

The models represent the architecture of the systems in terms of the structure of the system components and their interfaces, the mathematical or logical relations among them, their evolution over time and the rules that govern their design (Tolk and Hughes 2014). The simulation is then described as the execution of the system architecture models that is an essential element of model-based engineering which enables analyses in order to address measures of performance. As the system complexity increases, the simulation also becomes so complex that we ought to assess it not only as a tool for engineering a system but also a system that requires engineering. Accordingly, model-driven practices, namely, modeling, metamodeling and model transformations are perfectly applicable at various steps of engineering simulations for technical systems. Although there are many academic examples of model-driven practices in engineering of simulation systems in various domains, their industrial adoption in simulation of technical systems is not wide spread. Despite the fact that graphical modeling and model-to-text transformations are well-employed, metamodeling and model transformation practices have not become readily accessible. One of the root causes that hinder such accessibility is that current approaches derive from software engineering and are not compatible with the background of many in the simulation community. Formalisms that are endorsed with Meta Object Facility (MOF) (OMG 2015) by Object Management Group and model transformation approaches like Query/View/Transformation(QVT) (OMG 2016) are not natural to many simulationists with engineering background.

This paper investigates and discusses the applicability of the System Entity Structure and Model Base (SES/MB) framework for metamodeling and model transformations for simulations of technical systems. In the background, foremost, model-driven practices, namely modeling, metamodeling and model transformations will be revisited. Then, a short intensive review of the efforts to utilize metamodeling and model transformation in simulation will be presented. Finally, the SES/MB will be explained and it will be argued that SES/MB as a model-driven approach will facilitate applications of SES/MB for development of simulations in the technical systems domain. In the conclusion section, we will discuss the virtues of dissemination of these practices.

2 BACKGROUND

2.1 Model-Driven Approaches in Software Engineering

Model-driven methodology proposes the development of models and generation of executable software components through successive model transformations (Gasevic, Djuric and Devedic 2009). The key ingredients of model-driven methodologies are modelling languages, metamodels and transformations (Brambilla, Cabot and Wimmer 2012). Modelling languages enable the definition of a concrete representation for a model and metamodels are used to specify modelling languages. Transformations are described as the mappings between models which are specified at the metamodel level.

MOF is the basis of the model-driven integration framework of Object Management Group (OMG) for defining, manipulating, and integrating metadata and data (OMG 2015). It provides a meta-metamodel at the top level, which is called M3 layer of the four-layer metamodeling hierarchy. This meta-metamodel is then used to define more specific metamodels at M2 layer, such as the UML metamodel. The UML models that conform to the UML metamodel are at layer M1. Finally, the M0 layer includes the instances created from a UML model.

There are various categorization of model-driven approaches in software intensive systems. Following the description from Brambilla, Cabot and Wimmer (2012), Model-Driven Development (MDD) is a development paradigm that relies on models as primary artifacts and redefines implementation as (semi)automatic generation from the models using transformations. Model-Driven Architecture (MDA) is then the OMG's interpretation of MDD. And Model Driven Engineering (MDE) extends MDD to all engineering process areas. Model-Based Engineering (MBE) on the other hand is the pragmatic utilization of model-driven practices, not necessarily in an integrated fashion, in various steps of the engineering process. In this sense, all model-driven approaches can be categorized as model-based.

2.2 Model-Driven Approaches in Simulation

The research in the 1980's about methodologies for modeling and knowledge representation in simulation environments, such as in (Zeigler 1984, Ören, Zeigler, and Elzas1984), can be seen as roots for model-driven approaches in simulation. Employing model-driven methodologies in simulation in today's context was first discussed by Tolk (2002) when he proposed merging the concepts and ideas of High Level Architecture (HLA) into MDA. This position paper claimed that MDA should influence the future of modelling and simulation. Various efforts followed this claim and introduced employment of model-driven practices in MBE of simulation systems. A selection of works that introduced metamodeling and model transformations includes de Lara and Vangheluwe (2002) who presented $ATOM^3$, by which they supported metamodelling and explicit model transformations for model transformations between different formalisms. In 2006 Ozhan and Oguztuzun (2006) published their work in which they use metamodeling and model transformations for designing architectures for HLA-based federations based on a conceptual model. Based on the early works of the System Entity Structure (SES) and model base (MB) approach, Zeigler and Hammonds (2007) published an extended ontology for simulation-based data engineering, which was combined with the Discrete Event System Specification (DEVS) and implemented in Java in the MS4Me environment (Zeigler and Sarjoughian 2013). In 2008, Topcu et al. proposed the Federation Architecture Metamodel (FAMM) (Topcu, Adak and Oguztuzun 2008). Durak, Oguztuzun and Ider (2009) proposed their model driven simulation development approach in which they employ model transformations from OWL to UML (Ozdikis, Durak, and Oguztuzun 2009) and HLA Object Model (Ozdikis, Durak, and Oguztuzun 2010). Later Mittal and Douglass (2011) proposed the utilization of model-to-model transformations for generating DEVS Modeling Language from domain specific languages.

D'Ambrogio et al. (2010) came up with their MDD approach to develop DEVS simulations. It was one of the earliest attempts that targets a complete model driven simulation development pipeline from systems modelling to DEVS/JAVA code through successive (manual and automated) model-to-model and model-to-text transformations. In 2011, Cetinkaya, Verbraeck and Seck (2011) presented their MDD framework for modelling and simulation (MDD4MS) with which they focused on obtaining model continuity throughout this life cycle from problem definition to experimentation.

While previous efforts reported successful employment of metamodeling and model transformations, making these approaches available in widely used engineering simulation development environments such as MATLAB/Simulink or Scilab/Xcos were not in their scope. In recent efforts Legros et al. (2010) employ metamodeling and model transformations to automate the analysis and correction of MATLAB/Simulink models according to modeling guidelines. Later, Denil, Mosterman and Vangheluwe (2014) augmented MATLAB/Simulink, with rule-based model transformation capabilities for refactoring purposes. In 2015, Vanherpen et al. (2015) proposed model transformations for roundtrip engineering. Durak (2015a) discussed the accessibility of model transformation approaches to simulationists and proposed a transparent toolbox design for Scilab/Xcos, for in-place model-transformation.

Pawletta et al. (2014) proposed SES/MB as an ontology-assisted modeling approach in MATLAB/Simulink with providing a SES toolbox. It proposes an extension to SES/MB and provides an approach to compose models using the information from an SES. This effort is then used as a base line while developing a model-based testing methodology (Schmidt, Durak, Pawletta 2016). Further, Durak (2015b) presented an extension to Knowledge Discovery Metamodel based on SES for enabling reverse engineering of legacy simulations, and Durak et al. (2017) introduced an SES based simulation scenario development process.

It is important to mention feature modeling which is a comparable approach to SES/MB. Feature models are popular for variant specification (Kang et al. 1990) in software intensive systems domain. They are also well-applied to simulation models such as MATLAB/Simulink. Two examples from these efforts are Botterweck et al. (2009) and Haber et al. (2013). The application of SES/MB for variability management for MATLAB/Simulink, and in this sense comparing it with feature modeling approach was published by Pawletta et al. (2016).

Standing on the shoulders of previous SES/MB efforts, this paper will propose a MDD approach and further recommend SES/MB as an accessible approach for metamodeling and model transformations for technical systems simulation domain.

2.3 System Entity Structure

System Entity Structure (SES) is one of the many enhancements that has its roots from the system theory-based approach to modeling and simulation (Oren and Zeigler 2012). It is a high level ontology which was introduced for knowledge representation of decomposition, taxonomy and coupling of systems (Kim, Lee, Christensen and Zeigler 1990) and since then continuously developed.

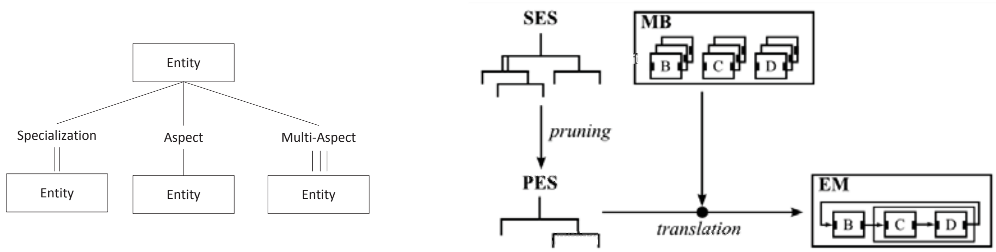

Figure 1: SES/MB Framework based on (Zeigler, Praehofer and Kim 2000).

Based on a clear, limited set of axioms, SES possesses four types of elements: Entity, Aspect, Specialization and Multi-Aspect. It can be represented as a directed and labelled tree composed of Entity and Aspect, Specialization and Multi-Aspect nodes. Entity is an object of interest. Variables can be attached to Entities. Aspect denotes the decomposition relationship of an entity while specialization represents its taxonomy. Aspects are represented by vertical single lines and Specializations by double lines. The Multiple-Aspect is a special kind of Aspect that represents a multiplicity relationship that specifies that the parent entity is a composition of multiple entities of the same type. Three vertical lines are used to represent Multiple-Aspect relation.

Uniformity, strict hierarchy, alternating mode, valid brothers, attached variables and inheritance are the axioms of SES (Zeigler 1984). Any two nodes with the same labels have isomorphic subtrees according to uniformity axiom. Strict hierarchy specifies a constraint that prevents a label from appearing more than once down any path of the tree. It is asserted with alternating mode axiom that, if a node is Entity, then the successor is either Aspect or Specialization, and vice versa. Valid brothers prevents two brothers from having the same label. With attached variables, it is enforced that variable types attached to the same item shall have distinct names. Inheritance axiom posits that Specialization inherits all variables and Aspects.

Pruning is an important concept in SES. It can be described as resolving the choices in Aspect, Multi-Aspect and Specialization relations and assigning values to the variables. Since multiple Aspect nodes may designate alternative decompositions of the system at the same hierarchical level, a particular subset can be chosen in pruning based on the modeler's purpose. Also possible variants of an entity are captured by a Specialization. Multiple Specialization nodes may designate families of alternatives in the same dimension, e.g., Size, Color, etc. in pruning, one entity needs to be selected from each Specialization (yielding e.g., Large_Red.). Pruning also specifies the cardinalities in Multi-Aspect relations. The outcome of pruning is called Pruned Entity Structure (PES), which is a selection-free tree.

The system entity structure and model base (SES/MB) framework combines the SES ontology with the classical workflow of modeling and the simulation of modular, hierarchical systems (Zeigler, Praehofer and Kim 2000). It introduces two general methods for an interactive or automatic generation of an executable simulation model. Basic models with a predefined input/output interface are organized in a model base (MB). The MB can be defined as a repository for basic models that describe dynamic behavior. Basic models are models of atomic or coupled systems, which can be composed by their input/output interface. Entity node attributes can be used to specify links to basic models in the MB. Pruning and translation, as pictured in Figure **1**, are proposed as a methodology for processing SESs and conducting model transformations. After pruning, a translation method is used to generate an executable simulation model (EM) based on the information of the PES and the referenced basic models from the MB.

3 SES/MB AS A MODEL-DRIVEN APPROACH

Model-driven methodology provides us means for model transformations between technical spaces. The technical space represents the context of the model. When we define simulation as the execution of the system architecture model, we can argue that the simulation development is transforming the system architecture constructed in system space to an executable model in simulation space. A model-driven approach is well applicable for such a task.

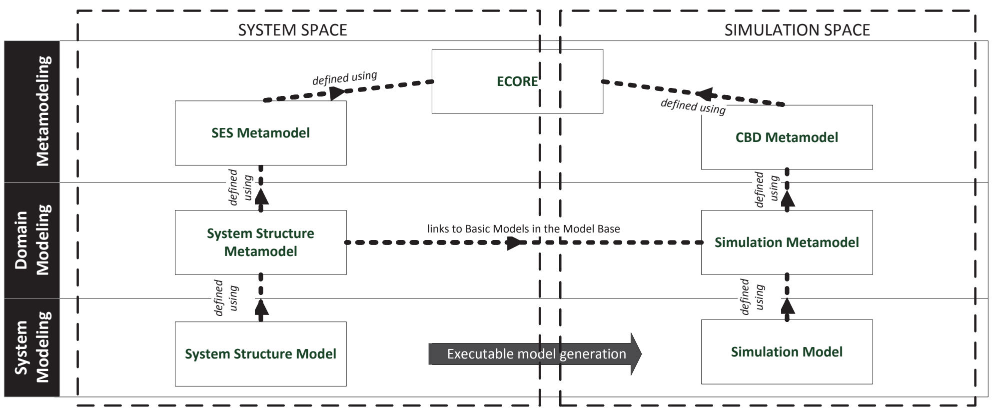

Figure 2: The SES-based MDD approach.

Figure **2** depicts the proposed SES-based MDD approach. Causal Block Diagrams (CBD) is a general formalism that is widely utilized for modeling of causal and continuous-time systems (Posse, De Lara and Vangheluwe 2002). They are extensively used in simulation of technical systems and supported by many simulation development environments, such as MATLAB/Simulink, as the basic modeling language. While we propose to use SES for modeling system architecture in system space, we would like to exploit CBD in simulation space for executable simulation modeling.

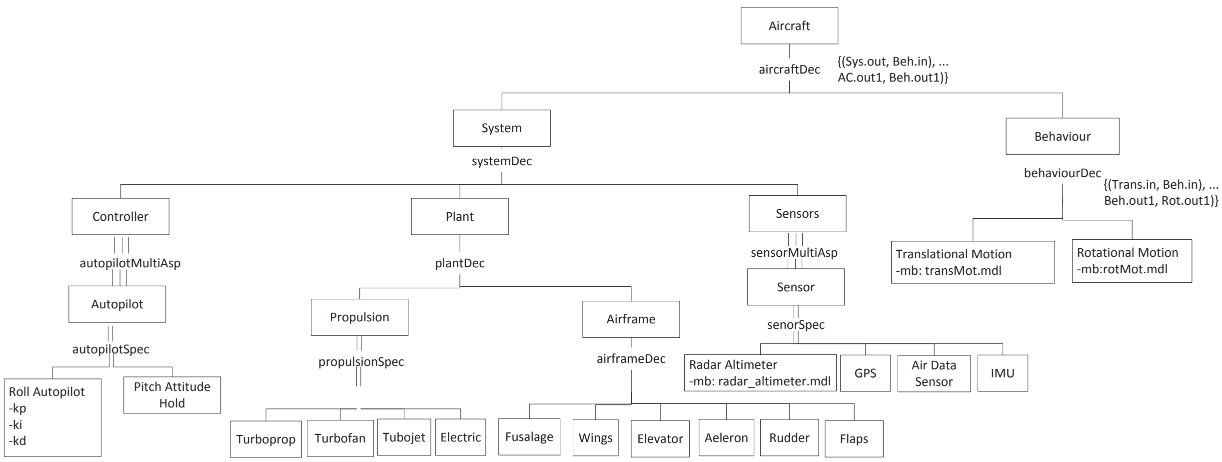

Figure 3: An excerpt from aircraft SES.

Adopting the layered framework of OMG, we propose a metamodeling layer, domain modeling layer and system modeling layer. In order to position the methodology in Eclipse Modeling Framework (EMF) ecosystem, SES metamodel is proposed to be developed using EMF CORE (ECORE). A simplified excerpt from such a metamodeling effort is presented in (Durak et al. 2017). Further, ECORE needs to be supported by a constraint language for expressing the constraints in SES axioms.

SES metamodel can then be used to specify the structure for a family of systems in domain modeling. This structure is regarded as a metamodel, namely System Structure Metamodel. Figure **3** presents an excerpt from an example aircraft system structure. This SES captures all possible variations of the system structure for the intended uses. For example, four different propulsion systems, namely Turboprob, Turbofan, Turbojet and Electric are captured using propulsionSpec specialization node. The identifiers that specify the corresponding basic model in the model base are captured with mb attributes at the leaf nodes. Examples can be seen in Translational Motion and Rotational Motion entities. System Structure Metamodel constitutes the entity parameters as well. Roll Autotpilot parameters are specified as an example in Figure **3**. The coupling relations among the entities, and accordingly among the corresponding basic models are captured as coupling parameter in aspect nodes. The reader can find examples in aircraftDec and behaviourDec nodes.

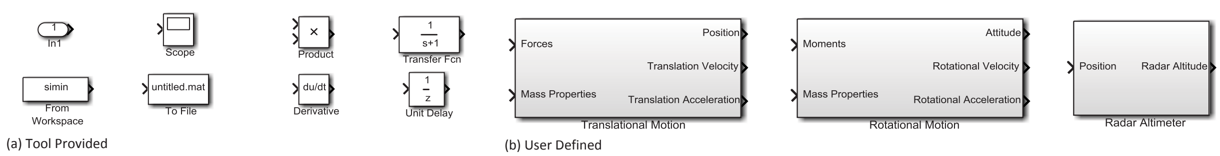

Figure 4: Simulation metamodel (Model Base).

In simulation space, metamodeling layer includes a CBD metamodel which is also developed using ECORE. Such a metamodel has been presented in (Denckla and Mosterman 2005). Domain modeling is then proposed as the model base construction. All executable model elements are defined using the CBD metamodel. They constitute the Simulation Metamodel. As presented in Figure **4**, the elements of Simulation Metamodel can be both tool provided elements like an integration block or a transfer function, or user defined blocks such as Translational Motion or Radar Altimeter. CBD based simulation modeling approaches, such as provided by MATLAB/Simulink or Scilab/Xcos have long been regarded as domain-specific languages (Botterweck et al. 2009). We can further extend this definition using embedded domain-specific languages which essentially inherit the infrastructure of another language and tailor it for a specific domain of interest (Hudak 1996). We can propose that user defined BMs specify an embedded domain-specific language for the particular domain of interest of some simulation study.

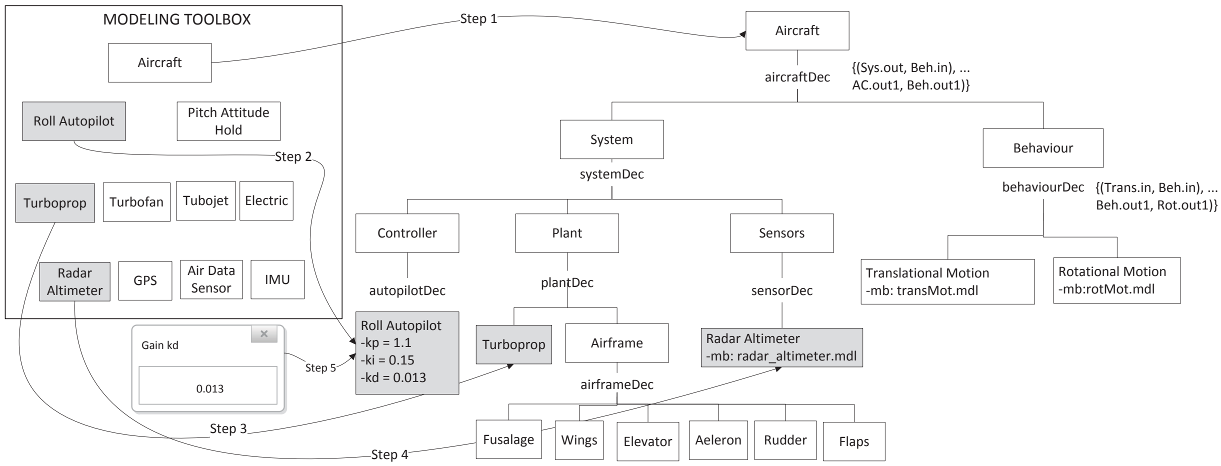

Figure 5: Interactive pruning for systems modeling.

The instances of the System Structure Metamodel appears in the System Structure Model in order to specify a particular system. The interactive pruning of the SES can be regarded as a systems modeling activity. PES is then proposed as System Structure Model. The decision nodes during the interactive pruning operation, such as the entity variants in specialization nodes becomes the elements in the modeling toolbox that is envisioned as the means of graphical system structure modeling. Figure **5** depicts proposed interactive pruning. The model parameters are also specified in this step. The reader can refer to Roll Autopilot parameters that specify a classical proportional, integral derivative controller gains as an example.

The executable model generation is the translation from PES to an executable model as proposed in SES/MB framework. We introduce it as the model transformation of the proposed MDD approach from System Structure Model to Simulation Model, or from system space to simulation space.

The interactive pruning process and executable model generation can be largely automated. The necessary extensions for the System Structure Metamodel are discussed in (Pawletta et al. 2014) and a first appropriate framework for the MATLAB/Simulink environment is introduced in (Schmidt et al. 2016).

4 CONCLUSION

Model-driven approaches are well studied in engineering of simulation systems, but the present paper is motivated by the lack of wide application of metamodeling and model transformations in simulation of technical systems. It points to the effect of low accessibility of the model-driven methodology on the diffusion of these practices, and proposes the System Entity Structure and Model Base framework as a means of metamodeling and model transformations for the simulation of technical systems. The approach is based upon the previous successful applications in MATLAB/Simulink for model-based testing and variability management, and proposes a model-based development approach.

The benefit of such an approach comes from the simplicity and adaptability of the SES/MB framework. It provides an accessible methodology for metamodeling, modeling and model transformation to simulationists with engineering background. Of course, a precondition is the availability of appropriate tools in engineers' simulation environments.

There is still a long list of future work in order to strongly support such a claim. This list includes a comprehensive metamodeling effort for System Entity Structure using a meta-metamodel supported by a constraint language such as OCL. Subsequently, proper mechanisms are required for introducing System Entity Structures in state of the simulation environments for domain specific metamodeling and system structure modeling to come up with streamlined workflows.

REFERENCES

Brambilla, M., J. Cabot, and M. Wimmer. 2012. *Model-driven Software Engineering in Practice*. Leipzig, Germany, Morgen & Claypool Publishers.

Botterweck, G., A. Polzer, A. and S. Kowalewski. 2009. "Using Higher-order Transformations to Derive Variability Mechanism for Embedded Systems". In *ACM/IEEE 12th International Conference on Model Driven Engineering Languages and Systems*, Denver, CO.

Cetinkaya, D., A. Verbraeck, and M. D. Seck. 2011. "MDD4MS: A Model Driven Development Framework for Modeling and Simulation". In *Summer Computer Simulation Conference*, The Hague, Netherlands.

D'Ambrogio, A. D. Gianni, J. L. Risco-Martin and A. Pieroni. 2010."A MDA-based Approach for the Development of DEVS/SOA Simulations". In *Spring Simulation Multiconference*, San Diego, CA.

De Lara J. and H. Vangheluwe. 2002. "Atom3: A Tool for Multi-formalism and Meta-modelling". In *5th International Conference on Fundamental Approaches to Software Engineering*, Grenoble, France.

Denckla, B. and P.J. Mosterman. 2005. "Formalizing Causal Block Diagrams for Modeling a Class of Hybrid Dynamic Systems". In *44th IEEE Conference on Decision and Control,* Seville, Spain.

Denil, J., P.J. Mosterman and H. Vangheluwe. 2014. "Rule-based Model Transformation for, and in Simulink". In *Symposium on Theory of Modeling & Simulation-DEVS Integrative*, Tampa, FL.

Durak, U., H. Oguztuzun and S.K. Ider. 2009. "Ontology-based Domain Engineering for Trajectory Simulation Reuse". *International Journal of Software Engineering and Knowledge Engineering*, vol.19, no.08, pp. 1109-1129.

Durak, U. 2015a. "Pragmatic Model Transformations for Refactoring in Scilab/Xcos". *International Journal of Modeling, Simulation, and Scientific Computing,* vol. 7, no. 1, p. 1541004.

Durak, U., 2015b. "Extending the Knowledge Discovery Metamodel for Architecture-driven Simulation Modernization". *Simulation*, vol.91, no.12, pp.1052-1067.

Durak, U., I. Pruter, T. Gerlach, S. Jafer, T. Pawletta, S. Hartmann. 2017. "Using System Entity Structures to Model the Elements of a Scenario in a Research Flight Simulator". In *AIAA Modeling and Simulation Technologies Conference*, Grapevine, TX.

Gasevic, D., D. Djuric and V. Devedic. 2009. *Model Driven Engineering and Ontology Development*. Dordrecht, The Netherlands, Springer-Verlag Berlin Heidelberg.

Haber, A., C. Kolassa, P. Manhart, P.M.S. Nazari, B. Rumpe, and I. Schaefer. 2013. "First-class Variability Modeling in MATLAB/Simulink". In *7th International Workshop on Variability Modelling of Software-intensive Systems*, Pisa, Italy.

Hubka, V. and W. Eder. 1998. *Theory of Technical Systems: A Total Concept Theory for Engineering Design*, Berlin, Germany, Springer-Verlag.

Hudak, P. 1996. "Building Domain-specific Embedded Languages". *ACM Computing Surveys (CSUR)*, vol.28, no.4es.

Jensen, J.C., D.H. Chang and E.A. Lee. 2011. "A model-based design methodology for cyber-physical systems". In *7th International Wireless Communications and Mobile Computing Conference*, Istanbul, Turkey.

Kang, K.C., S.G. Cohen, J.A. Hess, W.E. Novak, and A.S. Peterson. 1990. "Feature-oriented Domain Analysis (FODA) Feasibility Study". Technical Report CMU/SEI-90-TR-021, SEI, Carnegie Mellon University, Pittsburgh, PA.

Kim, T.G., C. Lee, E.R. Christensen and B.P. Zeigler. 1990. "System Entity Structuring and Model Base Management". *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 20, no. 5, pp. 1013-1025.

Lee, E. 2008. "Cyber Physical Systems: Design Challenges". In *11th IEEE International Symposium on Object Oriented Real-Time Distributed Computing (ISORC)*, Orlando, FL.

Legros, E., W. Schäfer, A. Schürr and I. Stürmer. 2010. "MATE-A Model Analysis and Transformation Environment for MATLAB Simulink". In *Model-Based Engineering of Embedded Real-Time Systems*, edited by H. Giese, G. Karsai, E. Lee, B. Rumpe and B. Schätz, pp.323-328, Heidelberg, Springer-Verlag.

Mittal, S. and S.A. Douglass. 2011. "From Domain Specific Languages to DEVS Components: Application to Cognitive M&S". In *Symposium on Theory of Modeling & Simulation: DEVS Integrative M&S Symposium*, Boston, MA.

OMG 2015. "Documents associated with Meta Object Facility™ (MOF™) Version 2.5" Available via http://www.omg.org/spec/MOF/2.5/. Accessed Nov. 02, 2016.

OMG 2016. "Documents associated with Meta Objet Facility (MOF) 2.0 Query/View/Transformation, v1.3" Available via http://www.omg.org/spec/QVT/1.3/. Accessed Nov. 02, 2016.

Ozdikis, O., U. Durak and H. Oguztuzun. 2009- "User-guided Transformations for Ontology Based Simulation Design". In *Summer Computer Simulation Conference*, Istanbul, Turkey.

Ozdikis, O., U. Durak and H. Oguztuzun. 2010. "Tool Support for Transformation from an OWL Ontology to an HLA Object Model". In *3rd International ICST Conference on Simulation Tools and Techniques*, Malaga, Spain.

Ozhan, G., and H. Oguztuzun. 2006. "Model-integrated Development of HLA-based Field Artillery Simulation". In *European Simulation Interoperability Workshop*, Stockholm, Sweeden.

Ören, T.I., B.P.Zeigler and M.S. Elzas. 1984. *Simulation and Model-Based Methodologies: An Integrative View*. Springer-Verlag, Berlin, Germany.

Ören, T.I. and B.P. Zeigler. 2012. "System Theoretic Foundations of Modeling and Simulation: A Historic Perspective and the Legacy of A Wayne Wymore". *Simulation*, vol. 88, no. 9, pp. 1033-1046.

Pawletta, T., D. Pascheka, A. Schmidt and S. Pawletta. 2014. "Ontology-assisted System Modeling and Simulation within MATLAB/Simulink". *SNE Simulation Notes Europe*, vol. 23, no.2.

Pawletta, T., A. Schmidt, B.P. Zeigler and U. Durak. 2016. "Extended Variability Modeling Using System Entity Structure Ontology within MATLAB/Simulink". In *the 49th Annual Simulation Symposium (ANSS '16)*. Pasadena, CA.

Posse, E., J. De Lara and H. Vangheluwe. 2002. "Processing Causal Block Diagrams with Graph grammars in Atom3". In *European Joint Conference on Theory and Practice of Software (ETAPS), Workshop on Applied Graph Transformation (AGT)*, Grenoble, France.

Schmidt, A., U. Durak, and T. Pawletta. 2016. "Model-based Testing Methodology Using System Entity Structures for MATLAB/Simulink Models". *Simulation*, vol.92, no. 8, pp.729-746.

Tolk, A. and T.K. Hughes. 2014. "Systems Engineering, Architecture, and Simulation" In *Modeling and Simulation-based Systems Engineering Handbook*, edited by D. Gianni, A. D'Ambrogio, and A. Tolk., pp. 11-41, Boca Raton, CRC Press.

Tolk, A. 2002. "Avoiding another green elephant - a proposal for the next generation HLA based on the model driven architecture". In *Fall Simulation Interoperability Workshop*, Orlando, FL.

Topcu, O., M. Adak, and H. Oguztuzun. 2008. "A Metamodel for Federation Architectures". *ACM Transactions on Modeling and Computer Simulation (TOMACS)*, vol. 18, no. 3.

Xiao, T. and W. Fan. 2012. "Modeling and Simulation Framework for Cyber Physical Systems". In *Advanced Methods, Techniques, and Applications in Modeling and Simulation*, edited by J.Kim, K. Lee, S. Tanaka, S. Park, pp. 105-115, Japan, Springer.

Vanherpen K., J. Denil, H. Vangheluwe and P. De Meulenaere. 2015. "Model Transformations for Round-trip Engineering in Control Deployment Co-design". In *Symposium on Theory of Modeling & Simulation: DEVS Integrative M&S Symposium*, San Diego, CA.

Zeigler, B.P. 1984. *Multifaceted Modelling and Discrete Event Simulation*, London, UK, Academic Press Professional Inc.

Zeigler, B.P., H. Praehoffer and T.G. Kim. 2000. *Theory of Modelling and Simulation*. 2nd ed. Orlando, FL: Elsevier Academic Press.

Zeigler, B.P., Hammonds, P.E. 2007. *Modeling and Simulation-Based Data Engineering.* Elsevier Academic Press.

Zeigler, B.P., Sarjoughian, H.S. 2013. *Guide to Modeling and Simulation of Systems of Systems.* London, Heidelberg Springer Pub.

AUTHOR BIOGRAPHIES

UMUT DURAK is a Research Scientist at Institute of Flight Systems of German Aerospace Center (DLR). He holds a PhD in Mechanical Engineering from Middle East Technical University in Turkey. His research interests include model-driven approaches applied to engineering of simulation systems and modeling and simulation based engineering of flight systems. His email address is umut.durak@dlr.de.

THORSTEN PAWLETTA is a Full Professor of Computer Science in the Faculty of Engineering at Wismar University of Applied Sciences. He holds a Ph.D. in Mechanical Engineering. His research interests include modeling and simulation theory, control of discrete event systems and tool development with focus on applications in engineering. His email address is thorsten.pawletta@hs-wismar.de.

HALIT OGUZTUZUN is a Professor in Department of Computer Engineering of Middle East Technical University. His research interests include distributed computing, programming languages and software engineering. His email address is oguztuzn@ceng.metu.edu.tr.

BERNARD P. ZEIGLER is an Emeritus Professor at the University of Arizona and Adjunct Research Professor in the C4I Center at George Mason University. He is known for inventing Discrete Event System Specification (DEVS). He is currently participating in RTSync Corp., a developer of the MS4 modeling and simulation software based on DEVS, as the Chief Scientist. His email address is zeigler@rtsync.com.