

# Merging OMG Standards in a General Modeling, Transformation, and Simulation Framework

Vitali Schneider, Anna Yupatova, Winfried Dulz, Reinhard German  
Friedrich-Alexander-University of Erlangen-Nürnberg  
Department of Computer Science 7  
Martensstr. 3, D-91058 Erlangen, Germany  
{vitali.schneider, anna.yupatova}@fau.de {dulz, german}@cs.fau.de

## ABSTRACT

Test-driven Agile Simulation (TAS) is a general-purpose approach that combines model-driven engineering, simulation, and testing techniques to improve overall quality for the development process. TAS focuses on the construction of system and test specification models that conform to the standards provided by the Object Management Group (OMG). Specifically, this approach aims at the detection of design errors by simulating the specified system and executing test cases as soon as possible at an early modeling level. In order to facilitate the development process we propose SimTAny: a versatile framework that enables seamless modeling, simulation, and testing of model specifications. The framework combines appropriate tools and software components within an integrated environment based on service-oriented architecture (SOA) and Eclipse RCP. The TAS approach as well as the SimTAny framework rely on various OMG standards and widely accepted tools. In particular, a combination of the UML and several standardized extension profiles namely SysML, MARTE, and UTP enables the development of high-quality software products based on a standard conform tool chain. The framework provides, among others, a MOFM2T standard conform model-to-text transformation component in order to generate executable simulation code for the simulation engine OMNeT++. In this paper we introduce the main features of the SimTAny framework with a special focus on the utilized OMG standards.

## Keywords

Test-driven agile simulation, model-driven engineering, model-driven testing, OMG, UML, SysML, MARTE, UTP, MOF2T

## 1. INTRODUCTION

Model-driven engineering techniques [1] are becoming increasingly important nowadays. By offering formal visual modeling languages, different abstraction levels, and relatively good tool support, such techniques became indispensable for engineering of large and complex systems. The stan-

dardization consortium OMG (*Object Management Group*)<sup>1</sup> provides various specifications, which cover different aspects of the model-driven engineering, grouped together under the concept of the *Model-Driven Architecture* (MDA) [2]. MDA references a number of related OMG standards like *Unified Modeling Language* (UML) [3], *Model to text transformation language* (MOFM2T) [4], or *MOF Query/Views/Transformation language* (QVT) [5]. Based on UML, standardized extension profiles are provided by OMG to enable modeling of specific issues. For instance, the *System Modeling Language* (SysML) [6] allows to model system specifications and requirements. Non-functional properties, time and analysis aspects can be expressed by the profile for *Modeling and Analysis of Real-time and Embedded systems* (MARTE) [7]. While the *UML Testing Profile* (UTP) [8] considers testing issues.

The approach of *Test-driven Agile Simulation* (TAS) [9] aims to provide an inexpensive and agile technique to improve the overall quality during the model-driven development process. Widely based on the standards provided by OMG, it combines model-based design and code generation techniques, along with model-based testing and simulation methods. The TAS approach enables the inspection of the involved models at early engineering stages among others by simulating the UML-based system model and by performing tests on it derived from the corresponding UML-based test model [10]. Using the model-to-text transformation methods standardized by MOFM2T the approach assist the transformation of UML-based specifications to executable simulation code. Due to the strict conformity with the OMG standards, the TAS approach aims to utilize popular and widely proven modeling techniques and tools.

In the next section we briefly outline the combined application of the OMG modeling standards for our TAS approach. In the subsequent section we focus on the OMG standards utilized inside of the provided SimTAny framework and introduce the main features of our tool environment.

## 2. MODELING WITH OMG STANDARDS

In the scope of the TAS approach we aim to provide a seamless standard-based modeling methodology applicable for various engineering aspects of the development process like requirements engineering, system design, analysis, and test. To achieve this, we suggest to merge the existing extension profiles standardized by the OMG group and based

<sup>1</sup><http://www.omg.org>

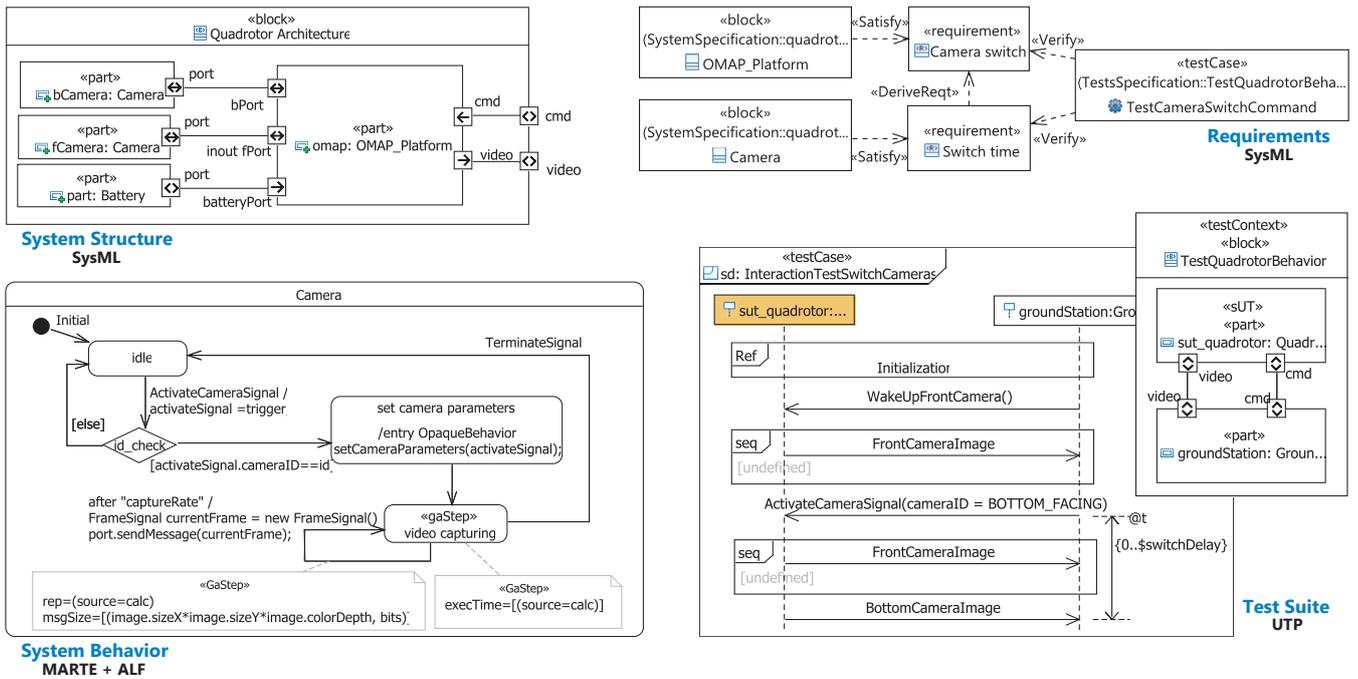


Figure 1: Modeling with OMG standards.

on its *Unified Modeling Language* UML, as illustrated in Figure 1. Although these profiles have been constructed for different application areas, they have some semantic and syntactic overlapping [11], that can be overcome by a suitable modeling methodology and selective application of their sub sets [12].

**UML.** At bottom, the proposed modeling methodology is based on UML. Designed as a general purpose modeling language, UML provides the basic elements for modeling of structures and behaviors. Due to the extension mechanism via profiles the modeling language can be adapted to specific needs. Thus, for instance, UML *classes* serve for definition of system *blocks* in SysML or of *test contexts* in UTP.

**SysML.** In our approach we support the SysML profile, which mainly provides an extension to UML for system engineering. Using *blocks*, *block definition diagrams*, and *internal block diagrams* of SysML the structure of a system may be defined in a way comfortable for system engineers. Also useful extension of SysML is *flow port* that adds a direction property to standard UML *ports*. Furthermore, SysML adds the possibility to define requirements directly in the models and to establish traceability to other model elements such as blocks, test or use cases through dependency associations.

**MARTE.** To add, for instance, non-functional properties, probabilities, times, resource and analysis aspects in UML models we apply several stereotypes of the MARTE profile. Thus, stereotypes of the MARTE's *Detailed Resource Modeling* sub-profile (DRM) can be utilized to specify the resource platform identifying hardware and software resources. Using *GaAnalysisContext* and *Var* stereotypes we specify alternative designs and parametrizable simulation experiments.

**ALF.** In order to achieve efficiency improvements in the modeling process we apply the OMG's standardized *Action Language for Foundational UML* (ALF) [13], which provides textual notation for detailed behavioral specifications. In contrast to a time-consuming and error-prone modeling with activity diagrams, ALF textual notations are more compact, intuitive and effective for most people. Such notations can be directly used, among others, to define behaviors of states and transitions in state machines or to define methods of operations. Furthermore, ALF comes along with a model library, that implies a variety of useful types and functions like collection types: *Queue*, *Set*, *Map* or like collection operators: *select* or *add*.

**UTP.** The UTP profile, which is also applied in our approach, offers several stereotypes to consider testing aspects such as *TestContext*, *TestComponent* and *TestCase*. Using this profile we can specify test objectives, configurations and concrete test cases to verify the behavior defined in the system model.

### 3. SIMTANY FRAMEWORK

Based on the Eclipse RCP<sup>2</sup> we are developing the SimTAny framework (formerly introduced in [9] as VeriTAS), which offers an integrated tool environment to provide automated support for the main tasks of our TAS approach. Among others, it combines tools and components for modeling, validation, model-to-simulation code transformation, simulation, and analysis. Due to the service-oriented architecture of SimTAny, several components can be easily adapted for integration in distributed and heterogeneous development environments [14].

<sup>2</sup>[http://wiki.eclipse.org/Rich\\_Client\\_Platform](http://wiki.eclipse.org/Rich_Client_Platform)

**Modeling Tool.** As modeling tool we currently use Papyrus<sup>3</sup>, since it is seamlessly integrated in Eclipse and provides a widely complete support for OMG standards. But other modeling tools, which can consistently export models into the OMG's interchange format XMI, can also be used. Our framework adds some extensions to Papyrus in order to improve the modeling efficiency. This primarily concerns behavior modeling with ALF textual editors for states and transitions.

**Model Transformation.** In order to automatically derive executable simulations from UML models, our framework offers a model-to-text transformation module. Implemented with the Acceleo<sup>4</sup> transformation framework, this module itself is based on the OMG standard model-to-text transformation language MOFM2T [4]. Currently we only provide the generation of executable code for simulation framework OMNeT++<sup>5</sup>. But the transformation module is designed to be extendable for different target simulation engines.

**Traceability.** Another popular OMG standard set of languages for model-to-model transformations, the MOF Query/View/Transformation (QVT) [5], is applied in our framework to derive traceability information. Using QVT we transform dependency associations between requirements, system and test elements from UML models to our own traceability model. Stored in a separate model, this traceability information can be easily analyzed and enriched with additional links to artifacts generated during model-to-code transformations, for instance. A special view is also provided by our framework to facilitate the traceability analysis.

**Simulation.** As mentioned before, our framework performs automatic transformation of UML models to the simulation code for OMNeT++. On the one hand, we use simulation to analyze the performance and behavior of the modeled system. On the other hand, we can execute modeled test cases on this simulation to achieve mutual validation of both the system and test models. Furthermore, the simulation code generated by our framework makes extensive use of Boost<sup>6</sup> C++ libraries, e.g. for efficient implementation of state machines in C++. Although Boost as a whole is not an official standard, but many of its libraries have been already included or are proposed to be included in the C++ standard.

**Analysis.** The resulting data like scalar and vector statistics achieved after a simulation run can be imported in our framework for further analysis. A number of predefined metrics can be calculated and plots can be created automatically for the selected data. Thereby, we apply the R Project<sup>7</sup> - the quasi-standard tool for statistical computing.

## 4. CONCLUSIONS

The standards provided by the OMG organization cover key aspects of the model-driven development process such as modeling, exchanging and transformation of models. To improve the overall quality in the development process we sug-

gest the approach of Test-driven Agile Simulation, which combines model-based engineering, simulation and testing techniques supported by the OMG standards. In this paper we offered an overview of the main features and standards applied in our framework SimTAny with the special focus to the OMG standards merged in our multidisciplinary modeling approach. Our ongoing efforts are focused, on the one hand, on the improvement of the suggested modeling methodology and of the consequential simulation code generation. On the other hand, we are currently evaluating our TAS approach in different application domains.

## 5. REFERENCES

- [1] S. Kent, "Model driven engineering," in *Integrated Formal Methods*, ser. Lecture Notes in Computer Science, M. Butler, L. Petre, and K. Sere, Eds. Springer Berlin Heidelberg, 2002, pp. 286–298.
- [2] Object Management Group (OMG), "MDA Model Driven Architecture Guide," 2003. [Online]. Available: <http://omg.org/cgi-bin/doc?omg/03-06-01>
- [3] —, "UML Unified Modeling Language," 2011. [Online]. Available: <http://omg.org/spec/UML>
- [4] —, "MOFM2T MOF Model to Text Transformation Language," 2008. [Online]. Available: <http://omg.org/spec/MOFM2T>
- [5] —, "MOF Query/View/Transformation (QVT)," 2011. [Online]. Available: <http://omg.org/spec/QVT>
- [6] —, "SysML Systems Modeling Language," 2012. [Online]. Available: <http://omg.org/spec/SysML>
- [7] —, "UML Profile for MARTE Modeling and Analysis of Real-Time and Embedded Systems," 2011. [Online]. Available: <http://omg.org/spec/MARTE>
- [8] —, "UTP: UML Testing Profile 1.2." [Online]. Available: <http://omg.org/spec/UTP>
- [9] A. Djanatljev, W. Dulz, R. German, and V. Schneider, "VeriTAS - A Versatile Modeling Environment for Test-driven Agile Simulation," in *Proc. of the 2014 Winter Simulation Conference*, Phoenix, AZ, USA, December 2011, pp. 3657–3666.
- [10] I. Dietrich, F. Dressler, W. Dulz, and R. German, "Validating UML Simulation Models with Model-Level Unit Tests," in *Proc. of the 3rd International ICST Conference on Simulation Tools and Techniques (SIMUTools '10)*, Malaga, Spain, March 2010.
- [11] H. Espinoza, D. Cancila, B. Selic, and S. Gérard, "Challenges in Combining SysML and MARTE for Model-based Design of Embedded Systems," in *Proc. of the 5th European Conference on Model Driven Architecture - Foundations and Applications*, 2009.
- [12] V. Schneider, A. Yumatova, W. Dulz, and R. German, "How to Avoid Model Interferences for Test-driven Agile Simulation based on Standardized UML Profiles," in *Proc. of the Symposium on Theory of Modeling and Simulation*, Tampa, USA, April 2014.
- [13] Object Management Group (OMG), "Action Language for Foundational UML (ALF)," 2013. [Online]. Available: <http://omg.org/spec/ALF/>
- [14] V. Schneider and R. German, "Integration of Test-driven Agile Simulation Approach in Service-oriented Tool Environment," in *Proc. of the 46th Annual Simulation Symposium (ANSS'13)*, San Diego, USA, April 2013.

<sup>3</sup><http://eclipse.org/papyrus>

<sup>4</sup><http://eclipse.org/acceleo>

<sup>5</sup><http://omnetpp.org>

<sup>6</sup><http://boost.org>

<sup>7</sup><http://r-project.org>