A metamodel approach to model driven service development

Yngve Lamo *
Bergen University College
Norway
yla@hib.no

Adrian Rutle
St. Francis Xavier University
Canada
arutle@stfx.ca

Model-driven engineering (MDE) is a branch of software engineering which aims at improving productivity, quality, and cost-effectiveness of software development by shifting the paradigm from code-centric to model-centric activities. MDE promotes models and modelling languages as the main artefacts of the development process and model transformation as the primary technique to generate (parts of) software systems out of models. Models enable developers to reason at a higher level of abstraction, while model transformation helps developers to avoid repetitive and error-prone tasks.

Models can be specified using general-purpose languages such as the Unified Modeling Language (UML) [3], but to fully realise the potential of MDE, models are often specified using domain-specific modelling languages (DSMLs) which are tailored to specific domains of concern. One way to define DSMLs in MDE is by specifying metamodels, which are models that describe the concepts and define the syntax of a DSML. A model is said to *conform to* a metamodel if each element in the model is typed by an element in the metamodel and, in addition, satisfies all constraints of the metamodel.

In this talk we will introduce a metamodel for the Simple Method Declaration Language (SMODL) [4]. SMODL is a lightweight approach to model driven development of services. From SMODL models one can generate service implementations in various programming languages such as JAVA and C#.

We use the Diagram Predicate Framework (DPF) [1] to formalise the SMODL metamodel. DPF provides a formalisation of (meta)modelling [5] and model transformation [6] based on category theory and graph transformations. DPF has a prototype implementation, the DPF Workbench [2], that supports the development of metamodelling hierarchies with an arbitrary number of metalevels. That is, each model at a metalevel can be used as a metamodel for the metalevel below. Moreover, the DPF Workbench checks the conformance of models to their metamodels by validating both typing and diagrammatic constraints. The DPF Workbench comes with a code generation facility that is used to generate program code from DPF specifications.

We will show an example of generating SMODL services from a DPF specification representing a part of a commercial offshore simulator system. Actually, in the future we want to build on the proposed solution to develop a service oriented architecture for the offshore simulator. Simulators are typically used for training purposes and thus it is important that offshore personnel train in an environment that is as near to their actual working environment as possible. This means that the simulator needs to simulate different ships and different configurations as realistic as possible. To configure the current simulator, a lot of manual development is needed and often the whole behavior of the ship needs to be reimplemented when certain details such as a ship's propeller type is changed. In this regard, the goal is to combine MDE and service orientation to automatically generate code which implements the behavior of the ships.

References

[1] Bergen University College and University of Bergen. *Diagram Predicate Framework Web Site*. http://dpf.hib.no/.

[2] Y. Lamo, X. Wang, F. Mantz, W. MacCaull, and A. Rutle. DPF Workbench: A Diagrammatic Multi-Layer Domain Specific (Meta-)Modelling Environment. *Studies in Computational Intelligence*, To appear.

[3] Object Management Group. *Unified Modeling Language Specification*, May 2010. http://www.omg.org/spec/UML/2.3/.

*This author is currently W.F. James Chair Professor at St. Francis Xavier University, Canada

[4] RUnit Software. *SMODL – Simple MethOd Declaration Language*. http://www.smodl.org/.

[5] A. Rutle, A. Rossini, Y. Lamo, and U. Wolter. A Diagrammatic Formalisation of MOF-Based Modelling Languages. In M. Oriol, B. Meyer, W. Aalst, J. Mylopoulos, M. Rosemann, M. Shaw, and C. Szyperski, editors, *TOOLS 2009*, volume 33 of *LNBIP*, pages 37–56. Springer, 2009.

[6] A. Rutle, A. Rossini, Y. Lamo, and U. Wolter. A formal approach to the specification and transformation of constraints in MDE. *JLAP*, 2012.