TECHNISCHE UNIVERSITÄT DRESDEN

Faculty of Computer Science Institute of Software and Multimedia Technology, Software Technology Group

SS2018 – Component-based Software Engineering

Composition Systems and Metamodeling

Professor: Prof. Dr. Uwe Aßmann
Tutor: Dr.-Ing. Thomas Kühn

Task 1 Composition Systems Basics

Composition systems simplify the development of large systems by focusing on loose coupling, separation of concerns, reuse, reducibility and standardization. This task repeats the terminology and the fundamental concepts of composition systems.

a) Clemens Szyperski provided one of the well-established definitions of a component [2]. How did he define a component? Try to summarize the key features of a component in your own words!
b) What are the three elements of composition systems? Try to explaint each part in your own words!
c) Is the UNIX Pipes and Filters approach a composition system? Explain the three parts!
d) Can LEGO™ be considered as composition system?[1] Explain your conclusion.

[1] http://www.lego.com/en-us/default.aspx

Task 2 Metamodeling Basics

Metamodeling is one central discipline in today's software engineering landscape. Especially for safety-critical systems, models are used to formally describe the structure and behavior of systems. Thus, certain properties can be proven. But not only for safety-critical systems, models are used. For example, one important engineering tool today are *Domain Specific Languages* (DSLs). This task repeats the basic terminology and concepts of metamodeling.

a) Explain the terms *Model*, *Metamodel* and *Metametamodel*. What are the relations across those elements?
b) The *Meta Object Facility* (MOF)[2] standard of the OMG emphasizes 4 layers of metamodeling (M3 - M0). In the lecture, we called this Meta-Pyramid. How are the terms of task a) aligned with those layers?
c) Why is there no 5^{th} layer M4?
d) What is a *Domain-Specific-Language* (DSL)? Give an example.
e) How can your example be aligned with the Meta-Pyramid?
f) Explain the terms *reflection*, *introspection* and *meta-object protocol* (MOP).
g) What happens when the MOP is changed?

[2] http://www.omg.org/mof/

Task 3 Metamodeling in Practice

We are going to design a composition system for classical components (according to Clements Szyperski [2]) utilizing the *Eclipse Modeling Framework* (EMF) [1] and `xText` language workbench. Components have a name, a set of attributes and a set of provided and required interfaces. Attributes have a name and a types (represented as a String). An interfaces has a name and a set of methods, whereas each method has a name, a return type (represented as a String) and a set of parameters. Like attributes, a parameter has a name and a type. Based on this model, compositions can be described as an instantiation and composition of components by connecting required and provided ports of the same type.

a) Download the latest version of the Eclipse Modeling Tools[3] and install the latest version of `xText`.[4]
b) Create an EMF-Metamodel (Ecore Model) for the composition system.
c) Create a DSL using `xText` that lets you design components and compose them in a composition language
d) Optionally, use the `xText` validation engine to validate rules that must be enforced in the static semantics of the language.[5]

References

[1] Dave Steinberg, Frank Budinsky, Ed Merks, and Marcelo Paternostro. *EMF: eclipse modeling framework*. Pearson Education, 2008.

[2] Clemens Szyperski, Jan Bosch, and Wolfgang Weck. Component-oriented programming. In *European Conference on Object-Oriented Programming*, pages 184–192. Springer, 1999.

[3] https://eclipse.org/downloads/packages/eclipse-modeling-tools/oxygen3a
[4] https://www.eclipse.org/Xtext/download.html
[5] Look at the `xText` help section for further information.