Component-Based Software Engineering Dipl.-Medieninf. Christian Piechnick INF 2078 `christian.piechnick@tu-dresden.de`	**Exercise Sheet No. 1** Software Technology Group Institute for SMT Department of Computer Science Technische Universität Dresden 01062 Dresden

Composition Systems and Metamodelling

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 tasks repeats the terminology and the fundamentals of composition systems.

1a) Clemens Szyperski provided one of the well-established definitions of a component [1]. How did he define a component? Try to summarize the key features of a component in your own words!

1b) What are the three elements of composition systems? Try to explaint each part in your own words!

1c) Is the UNIX Pipes and Filters approach a composition system? Explain the three parts!

1d) Can LEGO (`http://www.lego.com/en-us/default.aspx`) be considered as composition system? Explain your conclusion.

Bibliography

1. Clemens Szyperski, *Component Software: Beyond Object-Oriented Programming.* Addison-Wesley Longman Publishing Co., Inc., 2002 *The book is available in the SLUB.*

Task 2: Metamodelling Basics

Metamodelling is one central discipline in today's software engineering landscape. Especially for safety-critical systems, models are used to formally describe the structure and behaviuor 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 tasks repeats the basic terminology and concepts of metamodelling.

2a) Explain the terms *Model*, *Metamodel* and *Metametamodel*. What are the relations across those elements?

2b) The *Meta Object Facility (MOF)* standard of the OMG emphazises 4 layers of metamodelling (M3 - M0). In the lecture we called this *Meta-Pyramid*. How are the terms of task 2a) aligned with those layers?

2c) Why is there no 5$^{\text{th}}$ layer M4?

2d) What is a Domain-Specific-Language (DSL)? Give an example.

2e) How can your example be aligned with the Meta-Pyramid?

2f) Explain the terms *reflection*, *introspection* and *meta-object protocol (MOP)*.

2g) What happens when the MOP is changed?

Task 3: Metamodelling in Practice

We are going to design a composition system for classical components (according to Clements Szyperski). 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, whereby 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.

1. Download the latest version of the Eclipse Modeling Tools (`http://eclipse.org/downloads/`)
2. Install the latest version of xText (`http://eclipse.org/Xtext/`)

3a) Create an EMF-Metamodel (Ecore Model) for the composition system!

3b) Create a DSL using xText that let you design components and compose them in a composition language

3c) Use the xText validation engine to validate at least 3 rules that must be enforced in the static semantics of the language! (Look at the xText help section for further information)