Component-Based Software Engineering	**Exercise Sheet No. 1**
Dipl.-Medieninf. Christian Piechnick	Software Technology Group
INF 2080	Institute for SMT
`christian.piechnick@tu-dresden.de`	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) **Task:** 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) **Task:** What are the three elements of composition systems? Try to explaint each part in your own words!

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

1d) **Task:** 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) **Task:** Explain the terms *Model*, *Metamodel* and *Metametamodel*. What are the relations across those elements?

2b) **Task:** 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) **Task:** Why is there no 5^{th} layer M4?

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

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

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

2g) **Task:** What happens when the MOP is changed?

Task 3: Metamodelling in Practice

We are going to design a composition system for linear board games. The components are fields on the board, which have a color (red, white or black) and a shape (circle or rectangle). A board can be composed of fields by connecting the fields with a directed path. Each field has exactly one predecessor field and exactly one successor field. There are two special kinds of fields: a *start field*, which has no predecessor and an *end field*, which has no successor. The board must have exactly one start and exactly one end field. There must be a path from the start to the end field. Furthermore the board game can have at least two and at most 4 playing pieces. Each playing piece is assigned to exactly one field. At beginning of the game every playing piece is set to the start field.

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/`)
3. Install the latest version of Xpand (`https://eclipse.org/modeling/m2t/?project=xpand`)

3a) **Task:** Create an EMF-Metamodel (Ecore Model) for the board game structure!

3b) **Task:** Create a DSL using xText that let you design board games!

3c) **Task:** 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)

3d) **Task:** Use Xpand to generate HTML5 markup that visualizes your board game!