Design Patterns and Frameworks	**Exercise Sheet No. 9**
Dipl.-Inf. Steffen Zschaler	Software Engineering Group
INF 2097	Institute for Software and Multimedia Technology
`http://st.inf.tu-dresden.de/teaching/dpf`	Department of Computer Science
	Technische Universität Dresden
	01062 Dresden

Formal Models of Design Patterns II

Task 1: Composite in RBML

Read and understand [1]. This paper presents RBML, a UML-like notation for representing design patterns. The formal backing of this notation (representing the design patterns' role models as extension of the metamodel) allows for formal treatment of design patterns in actual models.

1a)

Use RBML to represent the Composite design pattern.

1b)

Go back to task sheet 3 on extensibility patterns and look at your solution for task 1a). Use the RBML techniques presented in [1] to show that this is indeed a realisation of Composite.

Bibliography

1. Robert France, Dae-Kyoo Kim, Sudipto Ghosh, Eungee Song, *A UML-Based Pattern Specification Technique.* IEEE Transactions on Software Engineering, Vol 30, number 3, pp 193-206, March 2004. *This paper is available online at the IEEE digital library by visiting http://ieeexplore.ieee.org/Xplore/DynWel.jsp and searching for it by title. You should have access to the digital library from any computer in the domain of the Computer Science Department.*

Task 2: OWL Observant

Read and understand [1]. This presents an approach that uses Semantic Web technology (in particular ontologies) to model design patterns. An ontology can be viewed (grossly simplifying) as a special kind of class diagram modelling concepts and their relations. An ontology, thus, provides vocabulary allowing to talk about a specific domain.

2a)

Use the technology from [1] to model the Observer design pattern.

Bibliography

1. Jens Dietrich and Chris Elgar. *A Formal Description of Design Patterns Using OWL.* In Proc. 2005 Australian Software Engineering Conference (ASWEC'05), IEEE Press, 2005.

Task 3: ρ-calculus and Template Class

Read and understand [1]. This presents a formal approach to writing down design patterns, which allows reasoning about the patterns. To understand it properly, you will also need to read, and understand the intuition of, [2], in particular Chapter 5, and Sections 6.1, 6.2, 6.6, 7.1, 7.2, 8.1, 8.5. Don't worry, though, it's fun reading!

3a)

Use the ρ-calculus to model the Template Hook meta-pattern. Template Hook essentially proclaims that there is a template method which invokes (i.e., depends on) a hook method. The two methods may be in the same class or they may not. Template Hook thus is essentially a role model.

3b)

Now use the ρ-calculus to model the Template Class design pattern. In this pattern there are two distinct classes, one for a template method and one for a hook method.

3c)

Can you represent Template Class using Template Hook? What does this mean?

Bibliography

1. Jason McC. Smith, and David Stotts. *Elemental Design Patterns – A Link Between Architecture and Object Semantics.* Technical Report TR02-011, March 2002, Dept. of Computer Science, Univ. of North Carolina, Chapel Hill.
2. Martin Abadi, and Luca Cardelli. *A Theory of Objects.* Monographs in Computer Science, Springer, 1996. *There are 3 exemplars in the library.*

Task 4: Discussion: Formal Representation of Design Patterns

From your experience with the pattern formalisations looked at so far, what are the benefits and drawbacks of attempts at formalising design patterns?