

Formal Models of Design Patterns I

In this exercise we look at some approaches to formal modelling of design patterns. We begin with simple role models and advance to UML-based models. In the next exercise we will then increasingly formalize our specifications.

Task 1: Role Modelling

You are to develop software for the management of horse shows.

In the world of horse shows, there are horses and riders who together obtain a starting number, with which they can inscribe for examinations. Of course, because the average horse show takes about two to three days, both horses and riders need a place to stay. It is one of the tasks of the organisation team to provide accommodation for horses and riders.

1a)

Develop role models for the situations described above using the notation from the lectures. Use at least two role models: One for riders and horses and their relations to examinations, and one for the management of accommodations.

1b)

Now map the roles from the previous subtask onto classes. Do you need to create new constraints? Are there alternative ways of mapping?

Task 2: Bridge Templates

We have seen in the lecture, that design pattern structures can be viewed as role models. In this task, we are going to look at some design patterns from this perspective.

2a)

Using the role-model notation from the lectures, provide a role model for `TEMPLATE CLASS` and one for `BRIDGE`.

2b)

Use the role identification notation to show how `BRIDGE` can be viewed as a specialisation of `TEMPLATE CLASS`.

Task 3: 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 meta-model) allows for formal treatment of design patterns in actual models.

3a)

Use RBML to represent the COMPOSITE design pattern.

3b)

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.*