Praxis der Software-Entwicklung

Development of a Campus Routing System

SS 2013

Institute of Theoretical Informatics
Prof. Dr. Dorothea Wagner

Software Design (Entwurf)

In the second phase of this course you have to create a software design of your system. The decisions you make during this phase have huge impact on the implementation. You should work very thoroughly since bad decisions may have extensive consequences.

Your software design should meet the typical quality characteristics you should know from the software engineering course, such as conformance (Konformanz), high cohesion (Kohäsion), low interconnection (Kopplung), secret principle (Geheimnisprinzip), principle of locality (Lokalitätsprinzip), and reusability of classes and subsystems.

Your software design should implement your functionality specifications document, where you specified mandatory as well as facultative criteria. Design your software in such a way that it contains all mandatory criteria and does not contradict to the facultative criteria. The decision to implement a facultative criteria should not result in a large reorganization of the software design.

The software design should also contain all interfaces to facultative criteria, a detailed design for the subsystems behind these interfaces is not required. However, it makes sense to figure out which facultative criteria you want to implement anyway and completely model these parts of your system. This saves time for the implementation phase as otherwise these extensions of the software design have to be done and documented in the implementation phase.

Submission

The final version of your software design has to be submitted as PDF document by 16.06.2013 (i.e., before 17.06.2013) via email to one of your advisors. Please note that since this document is part of the examination we *cannot* accept documents that are handed in even slightly too late.

Content

Your document should consist of at least 30 pages. We expect a detailed description of the architecture of your system. Please keep in mind that your software design document is supposed to be not only understandable for you and your advisors, but also by people that have only general knowledge about the field of software design. In particular, your document should contain at least the following items:

System Model Diagram (Architekturdiagramm). Shows the internal connection of the components in your system. You should use the diagram from you functional specification document and refine it appropriately.

UML Class Diagram. The class diagram contains the classes of your system and illustrates the relationships between them. Additionally, the class diagram must contain the most important methods and data-members. The exact signatures and helper methods are *not* required for this document.

Class Description. For each class, you should write a short description that contains the classes most important functionalities, as well as their data members. Moreover, you should illustrate the interaction between the different classes. For important classes you should have a more detailed description of the interaction than for less important classes.

Sequence Diagrams. For the most important procedures of your system, the design should contain sequence diagrams. The following cases should be covered.

- *System Initialization.* It should become clear what happens when starting the system. In particular which objects are created and who calls whom.
- *Mandatory Criteria.* You should cover all mandatory criteria. In particular your software design should contain a sequence diagram illustrating the simple case of computing a route between two points: The user selects a staring and an endpoint and starts the computation of a route.

For the cases where it is reasonable you should include sequence diagrams covering facultative criteria. You should focus on the important method calls, unimportant method calls can be shortened.

Miscellaneous. If necessary you can specify additional design details (e.g., using a database). You might extend your software design document with additional diagrams such as state transition diagrams.

Glossary. Like the functional sepcifications document your software design document should contain a glossary.

Colloquium

As in the first phase you have to defend your software design in a colloquium. The appointment for the colloquium will be arranged per team separately. The exact procedure of the colloquium will be announced in time.