Technical University of Denmark
DTU Informatics
Assoc. Prof. H. Baumeister

Spring 2014
Exercise Nr. 2
February 11, 2014

02291: System Integration

2.1 MUD Game Design

The task of this exercise is to create a design document containing sufficient information for a programmer to implement the system. The design document does not need to take any physical implementation into account.

- One should assume that each software component can talk to another component via a standard component protocol.
- The programmer himself chooses the right implementation language for each device and the communication technology between devices.

Furthermore, one should abstract away from any concrete user interface representation.

- Thus the user interface and its implementation is not part of the design to be described in this exercise. The focus should be on the application layer functionality.
- However, it may be helpful to draw sketches of the user interface to get a feeling for the functionality needed in the application layer.

Explain any non-trivial design decision that you make and state any assumptions that you make.[1] You can document them by

1. Describe the design issue / problem
2. Some alternative solutions and their consequences for the system
3. The selected solution

Note that the design you make should be justified by the use cases defined in the previous exercise. It does not make sense design something that is not justified by a requirement the user has. To reduce the size of the task, you should only make a design of your system such that the use case you have chosen to detail in the previous exercise form the basis of the design. You can detail more use cases if you think they lead you to an improved design.

2.2 MUD Game overview over the system architecture

Provide a high-level view of the architecture of the system using components, ports, and connectors using a composite structure diagram.

[1] You may want to check these assumption with me first before assuming them.

2.3 MUD Game rough system design

Use CRC cards to create a rough design of the system. Base your CRC cards session on the use cases you have detailed in the last exercise.

2.4 MUD Game Component Design

- Define the components of the system with their ports (i.e., a port is a set of required and provided interfaces).
- For each port describe the protocol using a state machine.

2.5 MUD Game Detailed Class Design

- Provide class diagrams to show how the components are implemented.
  - Some components can be implemented as a single class; other components can be implemented by a set of classes.
- Provide any necessary class invariants using OCL constraints.
- Specify the contract of each class diagram using OCL constraints (pre:/post:). Describe each class, their interaction, and the contract of each operation also using informal text.

2.6 MUD Game Behaviour Design

Describe the behaviour of each class using state machines.

2.7 Documenting the MUD game design

- Extend the report from last exercise with a section on the design of the MUD game. Thus the overall report should have the following structure:

  1. A title page
     1.1. Contains title of the report and authors
  2. Requirements
     2.1. Domain analysis
     2.2. Functional requirements
     2.3. Non-Functional requirements
  3, **Design**
     3.2. **Rough System Design** (contains the result of the CRC card session)
     3.2. **Component Design**
     3.3. **Detailed Class Design**
     3.4. **Behaviour Design**

- **Important:** Update your requirements documentation to include the knowledge gained when creating the design.

  - Note, this does not mean to include design decisions in your requirements model, but to include your better/more detailed understanding of the requirements as a consequence of doing the design.

- The PDF version of the paper should be submitted CampusNet.
  - The filename needs to be of the form sxxxxx_02.pdf (i.e. the student number of **one** of the group members, followed by an underscore, followed by 02, the number of the exercise with a prepended 0)