Model Based Software Engineering

RWTH Aachen Prof. Rumpe, WS 2012/2013 (18.02.2013)

Teaching assistant M.SC. Katrin Hölldobler

Annotation

This document is created from my memories directly after the exam and may contain minor faults or variances.

Task 1: Class Diagram and OCL (18+8 points)

a) Model a class diagram without using OCL that represents a reservation system for restaurants as follows:

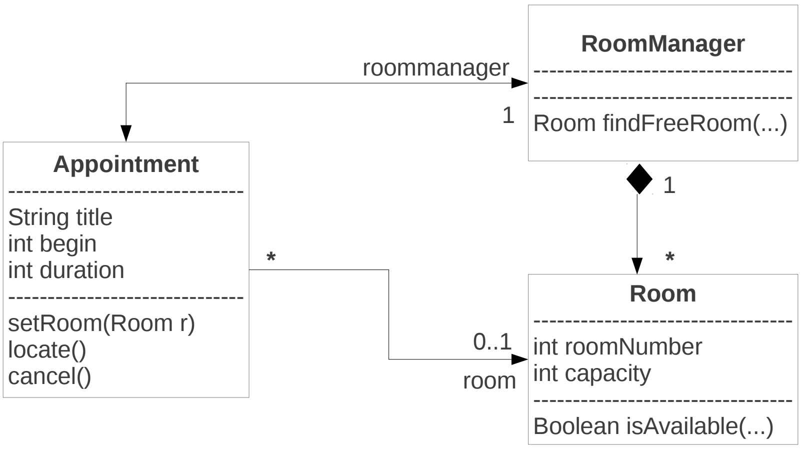
- A restaurant consists of seating areas
- Each seating area has one or multiple tables
- Seating areas may be indoor or outdoor areas, and they may be smoking or non-smoking sections
- each table has an identification number and a number of chairs
- A reservation of a table is made by a customer and has a starting timestamp (in seconds since 01.01.1970)
- A customer has a last name

b) Add OCL constraints to your class diagram that express the following:

- Each table has at least two chairs
- All indoor seating areas are non-smoking
- No two tables in a restaurant have the same id
- If there are multiple reservations of the same table, then the difference between their starting timestamps is at least two hours

Task 2: Sequence Diagram and OCL (24 points)

Consider a room managing system that is given by the following class diagram. Please read the description of this task carefully before you begin modeling.



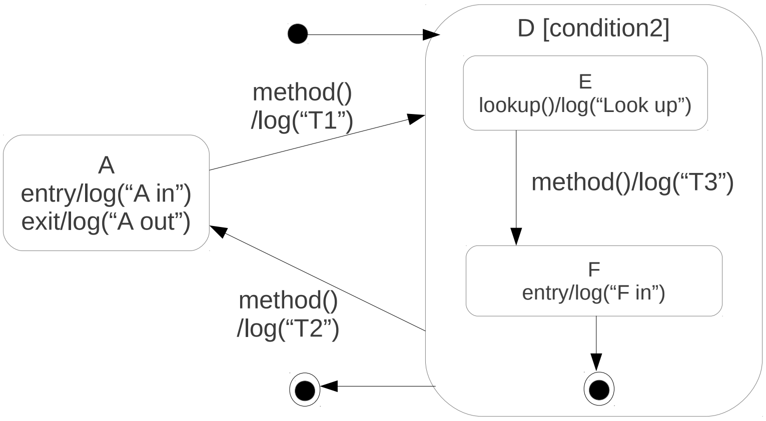
Create a sequence diagram for an execution of the *locate()* method of an *Appointment* object. Write all constraints that you use in OCL and use activity bars when objects are active. Model the execution of the method as follows:

- At the beginning of the method execution, there is no *Room* object linked to this *Appointment* object
- The *Appointment* object invokes the method *findFreeRoom* of a *RoomManager* object
- The *RoomManager* invokes the *isAvailable* method on a *Room r1*
- This method returns false
- The *RoomManager* invokes the *isAvailable* method on a *Room r2*
- This method returns true
- The *RoomManager* returns *r2*
- The *Appointment* object calls its own *setRoom* method with the free *room object*
- At the end of the method execution, *r2* is linked to the *Appointment* object

Hint: Parameters can be abbreviated by … unless they are part of the diagram.

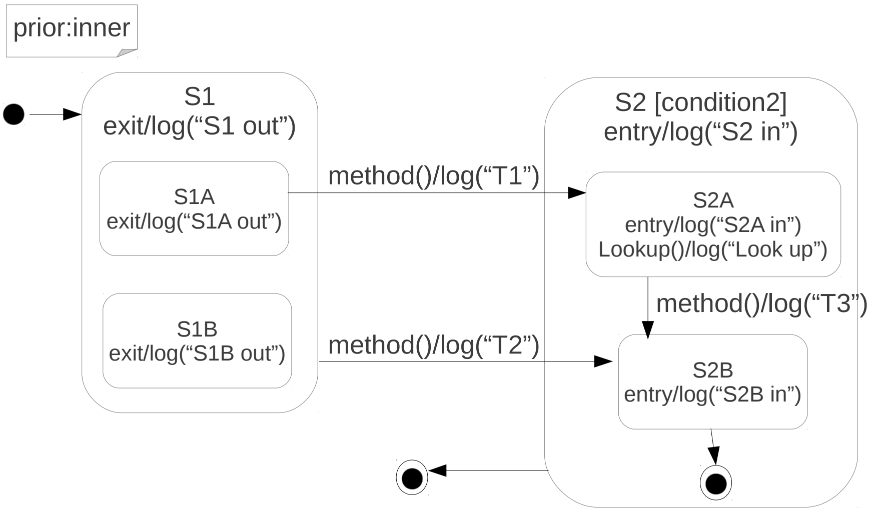
Task 3: Statechart – part 1 (20 points)

Simplify the following statechart according to the first 9 rules of the lecture. Please only draw the final result.



Task 4: Statechart – part 2 (10 points)

Use the following statechart for this task. Please read the description of this task carefully before you begin working on it.



Assume that we are in state *S1A* and receive the stimulus *method()* once. What is the resulting sequence of log messages?

Task 5: Questions (3+3+4 points)

1. Name the three main properties of a model as explained in the lecture

   (1) ______________________________

   (2) ______________________________

   (3) ______________________________

2. Define the following terms as in the lecture

   (1) Fault (German: Mangel)

   (2) Error (German: Fehler)

   (3) Omission (German: Auslassung)

3. Define Refactoring