

# INF5150 Obligatory Exercise, Autumn 2005

Version 051031

## ***Multiple Blind Date***

We are going to make a system to meet a group of people at a place that is optimally suited for those that will join.

We imagine that we are a third party vendor that offers a number of events identified by their type and time, e.g. "Go for a beer at 8 PM Saturday", "Play bridge at 2 PM today".

It is possible to join such an event by sending an SMS message to our dedicated number.

At appropriate time before the event, we will issue a message to the participant how he/she should use public transportation to reach a given meeting place.

We assume that the mobile phones sending the joining message can be positioned, and that we may use the Trafikanten system to find nearest bus stop, bus schedule and actual delays.

## **Drop 1**

- Make an appropriate set of UML 2.0 diagrams for a general Multiple Blind Date service.
- Apply at least class diagrams, composite structure diagrams and sequence diagrams
- Make some alternative solutions to the service where some parts of the specification are changed:
  - the time of the event is always NOW+30 minutes
  - the meeting place is given in the original event description
  - the meeting place shall be the most optimal meeting place for the participants
  - your own modifications
- Show how the modifications can be understood as refinements of the original service description by referring to the denotational semantics of sequence diagrams and the definition of refinement.
  - You need not describe each trace in full detail, but they must be described such that it is easy to understand what traces you are referring to.
  - This may be done by statements of the form "Let  $s_1$  be the set of all traces that we get from diagram  $d$  by choosing the first operand of the xalt-operator..."
- You may apply whatever tool you like for UML-modeling in Drop 1, but in Drop 2 we will require that you use Rational Software Modeler. Some of the tools that you may want to use for making sequence diagrams:
  - The SeDi plug-in for Eclipse
  - Rational Software Modeler
  - Visio with UML 2 template
- Drop 1 is to be delivered:
  - before 23.59 on October 14<sup>th</sup>
  - on mail to the teaching assistants and the lecturers

- consisting of *one* pdf-file including
    - Diagrams
    - Proofs
    - Proper explanations in English prose
- Drop 1 will be walked through on 21. October
  - First the group will present the report in Powerpoint of Pdf on video projector (10 minutes) The presentation shall be sent to Øystein Haugen the day before to be put on his machine.
  - Secondly another group will criticize the other group's report (5 minutes). The group will deliver this evaluation orally, but later send the critics on mail to the other group. Group 2 will criticize Group 1, group 3 criticizes group 2, etc. until Group 1 criticizes Group 7.
  - Thirdly, the assistants will give their evaluation, which is also written on one copy of the report.
  - Fourthly, the lecturers will give their evaluation concluded by a tentative mark (A- F). The mark has no formal impact on the students' final grades.

## Drop 2

- Create an executable UML design of the Drop 1 solution provided by the teachers.
  - The Drop 1 specification will be provided by the teachers directly after the walkthrough of the projects on 21. October.
- The executable UML design should contain the following:
  - UML state machines and composite structures (as well as corresponding class definitions)
  - SMS reception and transmitting through SMS mediators towards the PATS-lab (Telenor in Trondheim)
  - Positioning of mobile phones, also provided through the SMS mediators
  - Online observation of position of buses on Line 23 through Internet towards Trafikanten, provided through Trafikanten mediators
- The UML design shall be compiled onto JavaFrame through an RSM plugin.
  - This requires using some stereotypes from the provided JavaFrameProfile
  - JavaFrame is supplied in bytecode
  - The UML compiler is also provided as a plugin
- Explain how we may understand the design as a refinement of the Drop 1 specification (i.e. the solution provided by the teachers)
- Run the design and test it.
- Conduct a risk analysis using the CORAS tool
  - This involves filling in
    - Target of Evaluation Table
    - Value Definition Table
    - Asset Table
    - Risk Evaluation Criteria Table
    - HazOp Table
    - Consequence and Frequency Table
    - Risk Matrix

- This also involves
  - Defining the Risk Value Function
  - Carefully defining the target of analysis using UML and natural language (in particular, its border to the rest of the system) and the assumptions on which the analysis is based
  - Using the CORAS profile to specify treat scenarios for each risk that does not satisfy the risk evaluation criteria
  - Propose and specify treatments for these risks
- It is the responsibility of each group to define target and risk evaluation criteria in such a way that there are at least four risks (but not much more than four due to the workload) that does not satisfy the risk evaluation criteria.
- Drop 2 is to be delivered
  - before 18. November at 23.59
  - to the lecturers and the assistants on e-mail
  - consisting of
    - on pdf-file with diagrams, proofs and explanations
    - one executable “fat-jar” file that will execute the design in java
      - with a user explanation
    - one zip-file with the complete UML design