UNIVERSITETET I OSLO

Det matematisk-naturvitenskapelige fakultet

Exam in - INF5150 / INF9150
Day of exam: 3. December 2007
Exam hours: 09.00 – 12.00
This examination paper consists of 5 page(s).
Appendices:
Permitted materials: All written material.

Make sure that your copy of this examination paper is complete before answering.

NB: This exam text is only given in English since the course has been given in English this year. The candidate may, however, choose to answer in Bokmål or Nynorsk if he or she prefers.

A system for solving tasks

The context for this exam is the same as for the Obligatory Exercise 2 for INF5150 in 2008. TaskSolvers is software supporting communities set up to solve tasks in common. The software shall support several communities and be able to store a moderate amount of task-related data.

In this exam we only consider a small portion of the functionality. We consider only one community and we assume that the community has been set up.

In Figure 1 you will see the use of some of the following terms.

- *TTT* is the name of a task type ("Thing Take Time"). This task type has been already defined and may be requested and executed.
- *t1* is the unique name of one particular instantiation of TTT so that this instantiation can be easily identified by its users.
- *Sms*, *PosRequest* and *PosResult* refer to the signals of the SmsMediators package used in the obligatory exercise. We have skipped parameters for simplicity.
- *1720* designates the time of the *t1* instantiation
- *Oslo* designates the place of the *t1* instantiation

We assume that the proper algorithms to calculate distances from (say) Oslo to any of the users are easily available. You should not spend time on writing any such algorithms. You need only refer to algorithms by some informal explanatory text.

1 Modeling (35%)

In Figure 1 there is a sequence diagram specifying a scenario for the requesting a specific task and initiating it.

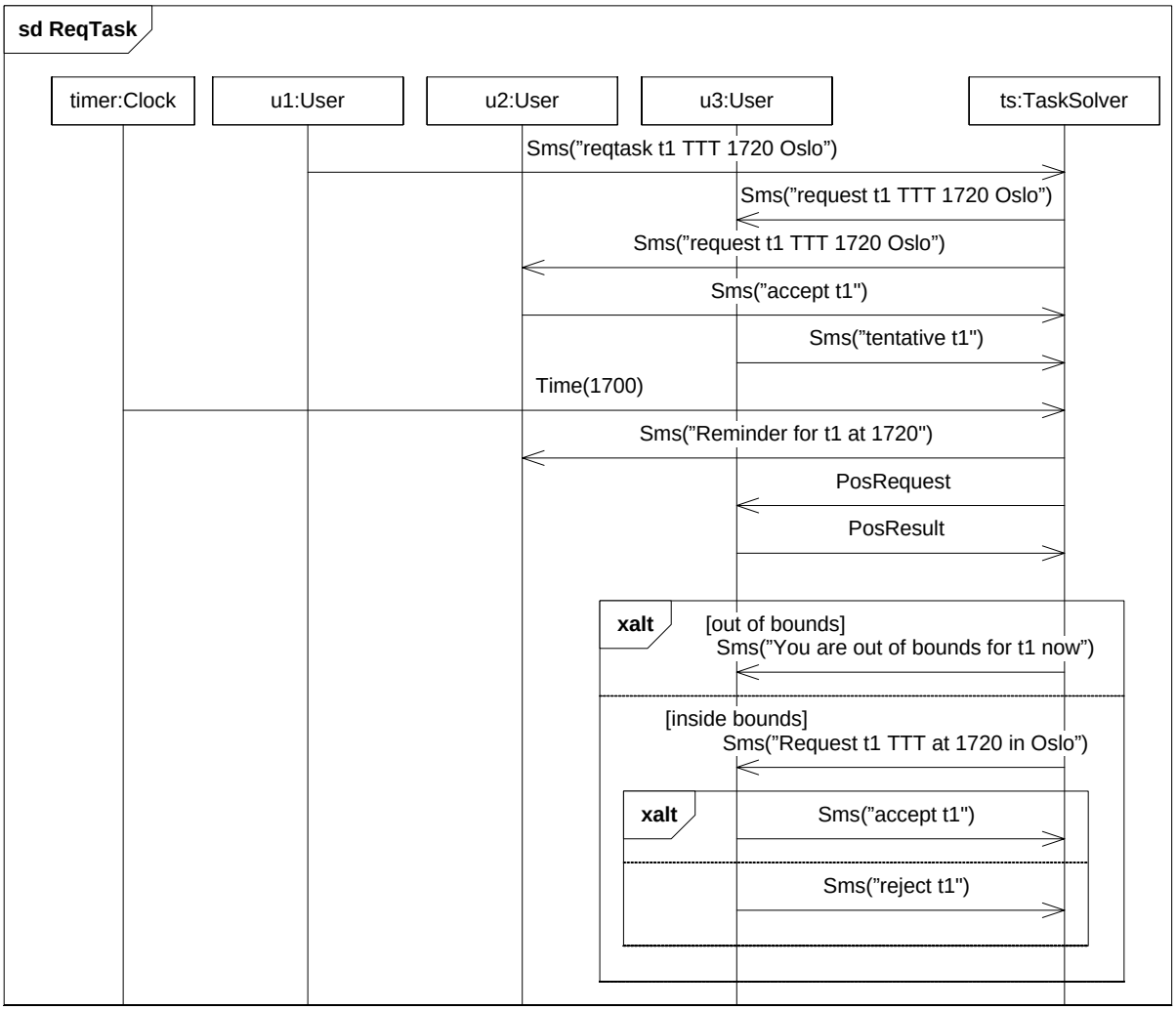

Figure 1Requesting a particular task and initiating it

The task type *TTT* is a task type that is already defined. When a particular instantiation *t1* is requested, the users of the community may accept, reject or respond tentative. The tentative responders receive special attention when the task is initiated. We define that tentative users that are out of bounds of the task location will not participate anyway, but those inside the bounds must give a definite response. We also assume that all those that at first accepted will participate and need only a reminder.

Notice that we have introduced a *Clock* that sends a message *Time* at *1700*, i.e. some minutes before the starting time of *t1*. We can imagine that the clock sends messages every minute, but that this particular message is the only one that we care about. You do not have to consider other *Time* messages.

1 a) Composite structure and internal signals

I. Define a composite structure of class *TaskSolver* based on the principles of architecture taught in INF5150 where you separate the archiving functionality, the routing functionality

and the service sessions. Please include multiplicities on the parts. Make sure that the Time signal is distributed to all sessions by including a multicast port.
Every *Sms* signal to *ts* should create a new session.

II. Define also necessary internal signals for communication with the archive and inter-session communication.

1 b) Decomposition

Define a decomposition of lifeline *ts:TaskSolver* for sequence diagram *ReqTask* given in Figure 1.

1 c) State machine

The definition of the sessions is a state machine type containing a number of submachinestates that are reached depending on the triggering signals. Figure 2 is a suggestion for how the state machine triggered by *Sms("reqtask ...")* could be defined.

I. Discuss whether the *reqtask* state machine in Figure 2 is consistent with your decomposition defined in 1 b).

II. Discuss what problems there might be with *reqtask* and how you would improve *reqtask*

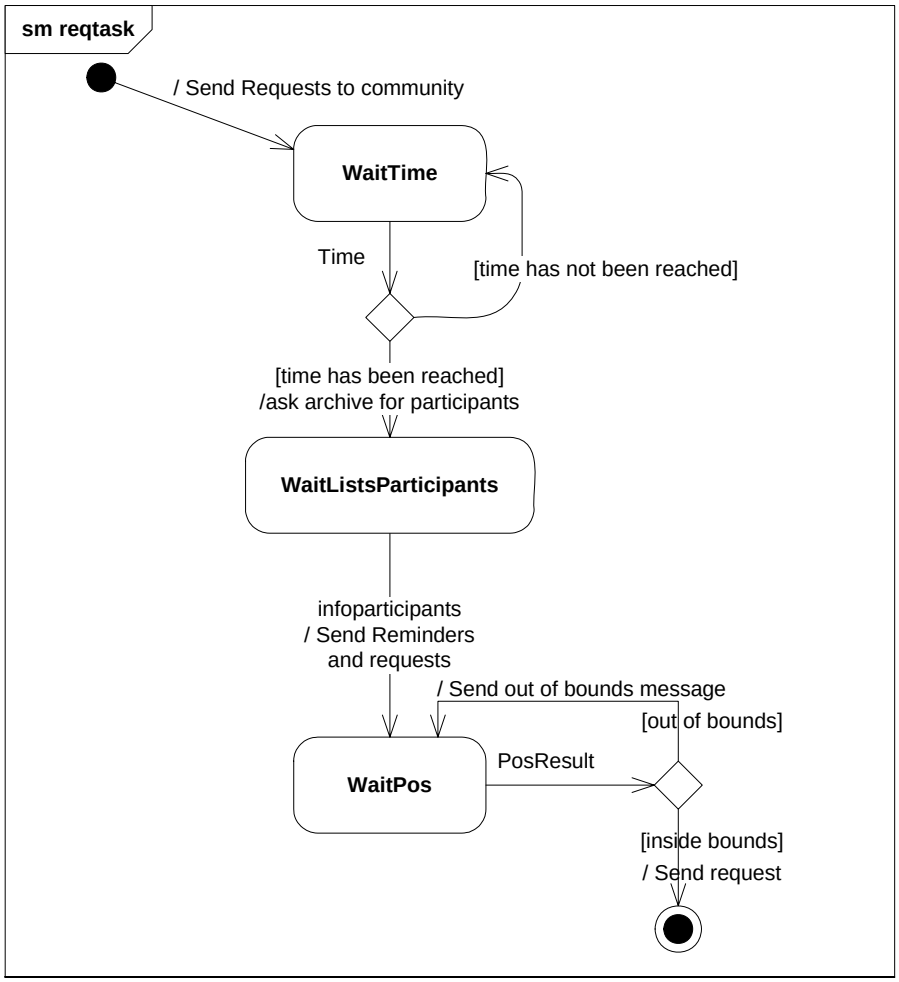

Figure 2 InitTask state machine type

"Send requests to community" will send Sms-es to all community members with the info of the task.
"ask archive for participants" means sending an internal signal to the archive to obtain info on the tentative and accepted participants.
"infoparticipants" is an internal signal containing two lists of telephone numbers (i.e. addresses of Sms-es): one for those who accepted and one for those who responded tentative. These lists are used to send the appropriate reminder Sms-es and PosRequest messages.
The two effects "Send out of bounds message" and "Send request" contain outputs of the appropriate Sms-es.

2 STAIRS (35%)

The exercises below refer to the sequence diagram of Figure 1, but do not depend on what you have answered on exercise 1 above.

2 a) Events

The STAIRS Tutorial associates two events with each message, a transmission event and a reception event.

I: How many events take place on the lifeline u2? Explain your answer.

II: What is (are) the first event(s) of Figure 1? (If there is more than one possibility, list all the possibilities.) Explain your answer.

III. What is (are) the last event(s) of Figure 1? Again explain your answer and if necessary list more than one event.

IV: The sequence diagram in Figure 1 is represented semantically by a set of interaction obligations. How many interaction obligations are there for Figure 1? Explain your answer.

2 b) Traces

I: Does the diagram in Figure 1 describe negative traces? Explain your answer.

II: Explain how the diagram in Figure 1 can be updated to describe exactly 3 positive traces by adding only message arrows.

2 c) Refinement

I: There are several simple ways to change the diagram in Figure 1 into a pure (in the meaning: does not involve supplementing) narrowing of the original diagram in Figure 1. Describe one and explain your answer.

II: Remove from Figure 1 the lifeline *u1* and the message sent from *u1* to *ts*. Is the resulting diagram a refinement of the original diagram in Figure 1? Explain your answer.

III: Obtain a new diagram from the diagram in Figure 1 by removing the guard [out of bounds].

(A) Is the new diagram a refinement of the original diagram in Figure 1?

(B) Is the original diagram in Figure 1 a refinement of the new diagram?

In both cases, explain your answer.

3. Risk Analysis (30%)

The exercises below refer to the sequence diagram of Figure 1, but do not depend on what you have answered on exercises 1 and 2 above.

3 a) Understanding

I: Assume the task solver software is considered for use in emergency situations and that you are hired to conduct a security risk analysis on behalf of the authorities. What would you identify as the most important assets? Identify at least two and motivate your answer.

II: Assume the emergency situation is an avalanche in the Norwegian mountains. Describe a
(a) non-human threat, and
(b) an accidental human threat
with respect to the task solver software and from the perspective of the authorities. Explain your answer.

III. Assume you have drawn a threat diagram for this emergency situation. With respect to STAIRS, would the traces corresponding to the threat diagram be positive, negative, inconclusive, or a combination? Explain your answer.

3 b) Risk modeling

I: Forget about the emergency situations. Assume instead that you are invited to become a member of a community. Hence, with respect to Figure 1, assume you are *u2* or *u3*. Assume you are invited by *u1* and are concerned that you may be victim of a scam.
Describe at least
(A) one asset,
(B) one deliberate threat,
(C) two threat scenarios,
(D) two vulnerabilities,
(E) two unwanted incidents
with respect to this situation on behalf of yourself.

II: Draw a threat diagram without likelihoods capturing this situation involving all the vertices described in I (A)-(E) above.

III: Assign likelihoods (i.e. frequencies/probabilities [different from 1.0]) to all relations and vertices so that the diagram is consistent under the assumption that the diagram is complete.