# INF5150 2008: Obligatory Exercise #2 Part 1 Version 080926

This obligatory exercise will be done in teams of 3-5 participants. The INF5150 teachers will select the project teams to try and distribute the competence evenly.

## *TaskSolvers – a task solving community using SMS and positioning*

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.

The INF5150 Exam from 2007 will serve as inspiration for this Obligatory Exercise. Some inspiration can also be found in "Challenges to UML 2 to describe FIPA Agent protocol" by Ø. Haugen (paper at ATOP – Agent-based Technologies and applications for enterprise interOPerability, Cascais, Portugal 2008).

### TaskSolvers dictionary

The following terms are important.

Task is some time-limited activity which is targeting a particular goal. The task is initiated by a member of a community and solved by that community in common. The initiator will also terminate the task – normally because it is completed.

Community is a group of people that have signed up to be included in that community. The community normally has some common denominator, but in fact only needs to have a name and a member roster. It should be possible also to remove oneself from the community.

Resource is some asset that may be needed to fulfill the task. Resources come in two variants, either static resources that are stored in a specific place always when not applied, and dynamic resources which need to be positioned when needed. The dynamic resources are normally tied to a person having a GSM phone. Special skills are modeled as resources.

Community Catalogue si a list of all community names and associated resources as well as member rosters.

The software should at least support the following services. The services are on purpose less specific than they could have been because the teams should make their own judgments and decisions.

### Set up a community

An initiator can set up a community just be giving a name and a short explanation. Initially the community will only have the initiator as member. The community is included in the community catalogue.

### Information about the community catalogue

Any person with a GSM phone may get the list of communities.

### Register as member of a community

Any person with a GSM phone may register as a member of a community.

### Unregister as member of a community

A person who is a member of a community can withdraw by sending an unregister message.

### Register a resource in a community

Any member of a community may register a resource of the community. If the resource is static, the idle position of that resource is considered to be where the registration has come from geographically. A resource is given a unique name and a short explanation.

### Register a task (type)

A member may register a task by giving a name, the number of persons needed and the resources necessary. This is a task type where neither the place of execution nor the time of execution is specified.

### Information of a community

Any member of the community can get information about the persons and resources of the community as well as its registered tasks.

### Requesting a task

This is the core service where tasks are actually executed. The request names the task type as well as the place and time of execution. If no time is given, the execution of the task is meant to be as soon as possible. The service should be similar to the agent protocol.
Every person in the community gets a request to participate and he/she should respond with their willingness. All dynamic resources should be asked for their participation and should respond. All communication goes via the TaskSolvers software. It is assumed that all participating persons and dynamic resources gather at the execution place on time. Some participants will get information to fetch static resources.

### Supplementing services

The teams should consider if there are other services that are needed or could be useful.

### Examples

Examples of communities may be “helping senior citizens”, “driving little league soccer team to matches”, “gather for political demonstrations”.
Examples of resources may be “lawn mower”, “car trailer”, “carpenter skills”, “large blank banners”.
Examples of task types could be “Lawn mowing, requiring 1 lawn mower and 1 person”, “Fetching stuff, requiring 1 car trailer and 2 people”, “Political Demonstration, requiring 2 large blank banners and 20 people”, “Driving to match, requiring 3 cars and 3 people”.

## *Technicalities of Oblig 2*

### Delivery

- Absolute deadline!
- A report in pdf sufficient for the evaluators to read to understand the system
    - The pdf-file shall have the following naming convention:
      **G*n*-Oblig2.pdf**
      where *n* is your group number, and be placed on top level of your group area
    - The report shall be sufficient to understand your system in detail. The report shall also contain the security analysis including the CORAS diagrams.

- The di2-file and the uml-file (together the full UML model)
    - The di2- and uml-files shall have the following naming convention:
      **G*n*-Oblig2.di2 and G*n*-Oblig2.uml**
      where *n* is your group number, and be placed on top level of your group area

## Peer-review

- Each project group will be reviewed by another project group
- The evaluating project group will produce a *test specification* with UML Testing Profile for the evaluated project.

## Demonstration

- We shall during the final session of the course run the projects
- We may try and run several projects in parallel
- The test specifications will be executed by users that are neither designers nor peer-reviewers

## Evaluation

- The lecturers will grade each project and give their qualitative evaluation
- Some individuals may be selected for individual examination of their work in Oblig2. Individuals may be failed even when the whole group passes.