**Exam Behavioural Dynamics 2008** (December 18, 2008, 12.00-14.45, M143).

**Part 1 (45 pts)**

In this assignment the following case is analyzed:

Agent A explores its environment with the aim to identify friendly agents. Agents possess some valuable items (e.g., food), which they may share with other agents. It is assumed that agents can observe the amount of each other's items and can communicate with each other.
Agent A identifies friendly agents as follows: when agent A has a low amount of items, it requests items from an agent with a high amount of items. If this agent refuses to provide the requested items, it is considered as an unrelated agent. If it provides items for at least two A's requests in a row, then it is considered as a friendly agent. When agent A has plenty of items, it provides items upon request to all agents except for unrelated ones.

Only the dynamics of agent A (and not of the environment) are required to be modelled in this assignment.

Assume the following relevant state properties for the example:

Input state properties

| | |
|---|---|
| obs(items_provided_by(agent)) | A observes that the requested items are provided by agent |
| obs(items_rejected_by(agent)) | A observes that the requested items are rejected by agent |
| obs(low_items(agent)) | A observes that agent has a low amount of items |
| obs(plenty_items(agent)) | A observes that agent has plenty items |
| obs(items_requested_by(agent)) | A observes that items have been requested by agent |

Output state properties

| | |
|---|---|
| performed(request_items_from(agent)) | A requested items from agent with plenty of items |
| performed(provide_items_to(agent)) | A provides items to agent with a low amount of items |

Internal state properties

| | |
|---|---|
| belief(potential_friendly(agent)) | A believes that agent is potentially friendly |
| belief(friendly(agent)) | A believes that agent is friendly |
| belief(unrelated(agent)) | A considers agent as unrelated |
| belief(low_items(itself)) | A believes that it has a low amount of items |
| belief(plenty_items(itself)) | A believes that it has plenty of items |
| belief(low_items(agent)) | A believes that another agent has a low amount of items |
| belief(plenty_items(agent)) | A believes that another agent has plenty of items |

a) Give an example of a trace (showing input, output and internal state properties), which describes the behaviour of agent A in case it encounters another agent, which is identified first as an unrelated agent, but later as a friendly agent. **(6)**

b) Show the dynamics of the example in graphical form. Do not forget to indicate which state properties are persistent. **(9)**

c) Write down at least 3 *executable dynamic properties* that characterise these dynamics. **(5)**

d) For the properties you defined in c), indicate which ones are *step properties* and which ones are *persistence properties*. **(5)**

e) For one of the internal state properties, indicate by which of the dynamic properties its *functional role* is defined. **(5)**

f) Give a set of dynamic properties that specifies the input-output correlation from an **external** perspective. **(5)**

The following two questions are independent from the case study:

g) Explain in your own words why it is important to specify requirements for overall system behaviour when you build an Agent System. In your answer, make use of the term *requirements refinement*. **(5)**

h) Give an example of a *reasoning trace* $\gamma$ for which the following property fails:

**GP2 Correctness of rejection**

Everything that has been rejected does not hold in the world situation.

$\forall t{:}T\ \forall A{:}INFO\_ELEMENT\ \forall S{:}SIGN$
$state(\gamma, t) \models rejected(A, S)$
$\Rightarrow state(\gamma, t) \models not\ holds\_in\_world(A, S)$

Explain why this property fails. **(5)**

**Part 2 (45 pts)**

One of the problems often encountered by the government is that results obtained from theoretical research at universities are not immediately applicable in practical settings. Therefore, institutes such as TNO (Netherlands Organization for Applied Scientific Research) have been created. Such an organization can be briefly described as follows.

There are three groups within the organization. First of all, there is the Client Group. In this group, a representative of TNO (abbreviated to TNO-Rep) is present, and a Client, that has a certain problem they want TNO to solve. The agent playing the TNO-Rep role also plays a role within the second group, namely the TNO Group. Here, the agent plays the role of a Project Leader, and interacts with a TNO Researcher. Finally, in the third group, interaction between TNO and the University takes place, called the Knowledge Transfer Group (KT-Group). Herein, a University Researcher (UR) participates, who interacts with the TNO Researcher Representative (RR). The RR is in fact played by the same agent as the Researcher within TNO, and forms the bridge between TNO and the University.

The dynamics of the organization are specified as follows. The whole process is started when the Client role outputs a request to the TNO Representative in the Client Group concerning a solution to be found to a specific problem. After the TNO Representative has received this problem, the Project Leader outputs the problem to the TNO Researcher in the TNO Group. Once the TNO Researcher has received this information, the TNO Research Representative outputs a request for relevant theoretical research to the University Researcher within the Knowledge Transfer Group. The University Researcher as a result, provides the TNO Research Representative with the appropriate theoretical results. After the TNO Research Representative has received this information, the TNO Researcher within the TNO Group outputs the solution to the problem based upon the theoretical results. This solution is received by the Project Leader within the TNO Group, and as a result outputted by the TNO Representative to the Client in the Client Group. Eventually, the Client receives this information.

a) Express the AGR-specification of this organization in **graphical** format **(15)**

b) Express the behavior of the organization in terms of semi-formal dynamic properties. Try to limit yourself to the behavior described in the text above. **(15)**

c) Provide a proof tree for the organizational property *" if the Client outputs a request for a solution to a particular problem, then the Client eventually receives a solution "* **(15)**