Exam for IN1905-I Kennistechnologie

31 March 2008

This exam will test your knowledge and understanding of the material provided to you and presented in the lectures and the book of Lin Padgham and Michael Winikoff, *Developing Intelligent Agent Systems* as well as the notes of Paul Brna, *Prolog Programming: A First Course*. Using the book of Padgham and Winikoff during the examination is allowed. It is *not* allowed to use other materials such as slides or the notes of Brna during the exam. You will have 3 hours (from 9 till 12) to complete the exam. It has 4 questions, for a total of 100 points. Please don't include irrelevant information: you will be marked down for this. Before you hand in your answers, please check that you have put your name and student number on top of every sheet you hand in.

Questions

Question 1 *15 points*

(a) (10 points) A multi-agent system is usually very *dynamic*. Explain why this is the case.

(b) (5 points) Planning in highly dynamic environments is often not realistic since the environment is too unpredictable. Explain why dynamic environments are typically also *unpredictable*.

> **Solution:** Cf. slides 12-14 from Lecture 1, and page 2 and Section 2.3 in *Developing Intelligent Agent Systems*.

Question 2 *50 points*

In this assignment we consider a system called the Automated Customs Environment, known as ACE. This system automates processes related to Customs and Border Protection (CBP). ACE is a comprehensive system to facilitate CBPs operations related to international trade management. One of the more important tasks of the system is to (pre)process submitted entries or requests to import products into the country. Over 90% of entries are submitted through the Internet to Customs. At the core of the system is a database, the International Trade Data System (ITDS), which holds data required for trade purposes for all government entities with requirements that affect trade movements. One of the main aims of the system is to minimize transportation delay as a result of customs procedures while at the same time ensuring security of main ports. The ACE system enables the following functionalities:

- Customs Import Assessments: import requests are reviewed and assessed by the system, and checked against regulations incorporated into the Trade Act, the Bioterrorism Act and other recent legislation intended to secure the country.
- Customs Release Management: imports are released and importers are granted rights to import shipments; details of the release are established, such as the port of entry used as well as mode of transportation (e.g. rail, air, vessel). Importers are informed automatically about the decision and associated rights and obligations.
- Importer Account Summary Statement: Importers will be able to make a monthly accounting for their import activities, and are allowed to make payments of duty each month for all shipment released during the previous month. Payment management functions such as regular payments, as well as outstanding fee handling have been automated. Importers with outstanding fees are not granted rights to import shipments.
- International Trade Data System: One of the functions of the system is to maintain and provide data about imports to relevant govermental bodies. One database is maintained that will supply data to all government entities that need information on international trade movements.

In this assignment you will specify the goals, actions and percepts, a scenario, decompose the system into agents, and design one protocol.

(a) (10 points) Identify the system goals and provide a system goals diagram. Your system goals diagram should minimally have 10 goals.

(b) (5 points) Identify the actions and percepts relevant for agent system design. You should aim to be as complete as possible. Provide a short description of each action and percept.

(c) (10 points) Develop a *detailed* scenario (based on the information provided above) for customs release management. Use a request for importing as the trigger of the scenario and include the granting of rights to import (if applicable).

(d) (15 points) Decompose the system into agents and draw an agent acquantaince diagram. Your system should consist of at least 4 agents. Provide a descriptor for each agent including (i) name, (ii) short description, (iii) percepts, (iv) actions, (v) uses data, (vi) produces data, and (vii) goals.

(e) (10 points) Draw a protocol that covers the exchange of messages related to customs release management. The protocol should be exhaustive and cover all possibilities.

Question 3 *25 points*

This assignment concerns the definition of two Prolog procedures.

(a) (10 points) Define a procedure `del1st(X,Y,Z)` which returns in `Z` the list obtained from list `Y` by removing the first occurrence of element `X`, if present; otherwise the predicate `del1st(X,Y,Z)` should fail.

> **Solution:**
> Cf. `http://computing.unn.ac.uk/staff/cgpb4/prologbook/node219.html`.

(b) (15 points) This assignment asks you to write a predicate that can help animal psychologists to analyze the communication between cat and dog. We assume that their respective vocabulary is defined by the following set of Prolog facts.

```
dogsound(wouf).
dogsound(wrrouf).
dogsound(grrrr).
dogsound(grr).
dogsound(aoaouuuuuu).
dogsound(whimp).
catsound(miaw).
catsound(miiiiiiaaauuuuuw).
catsound(scratch).
catsound(fshhhhhhhhtt).
```

This question concerns the power game between cats and dogs. You should write a predicate, `winner/2`, that determines who ended up being the strongest in a cat and dog dialogue. We say that the dog is the winner if it gets the last word, unless this is `aoaouuuuuu` or `whimp`; otherwise the cat is the winner. Example:

```
?- winner([miaw,wouf,wouf,wrrouf,grr,fshhhhhhhhtt,scratch,whimp,whimp],W).
W = cat ? ;
no
```

Question 4 *10 points*

This question concerns the agent programming language GOAL.

(a) (10 points) Explain which actions the following GOAL agent may perform *next*, given its belief and goal base as listed below.

```
:main: blocksWorldAgent
{
  :beliefs{
     block(a), block(b), block(c), block(d), block(e), block(f).
     on(c,a), on(a,table), on(b,table), on(d,b), on(e,d), on(f,table).
     clear(X) :- (block(X), not(on(Y,X))).
     clear(table)
   }
   :goals{
     on(a,b), on(b,c), on(c,table), on(d,table), on(e,d), on(f,e)
   }
   :program{
     if bel(on(X,Y), clear(X), clear(Z)), goal(on(X,Z))
                  then putOn(X,Z).
     if bel(on(X,Y), not(clear(X))), goal(on(X,Z))
                  then adopt(clear(X)).
     if goal(on(X,Z)), bel(on(X,Y), not(clear(Z)))
                  then adopt(clear(Z)).
     if bel(on(X,Y), clear(X)), goal(clear(Y))
                  then putOn(X,table).
     if bel(on(X,Y), not(clear(X))), goal(clear(Y))
                  then adopt(clear(X)).
     }
   :actionspec{
      putOn(X,Z) {
        :pre{ clear(X), clear(Z), on(X,Y) }
        :post{ not(on(X,Y)), on(X,Z) }
      }
   }
}
```

Solution: The agent can perform the following actions:

- `putOn(c,table)` because block `c` is misplaced, clear and should be on the table according to the goals of the agent; the first action rule in the program section allows the agent to perform this action next.

- `adopt(clear(a))`, `adopt(clear(b))` and `adopt(clear(d))` are enabled by the second as well as the third rule, since each of these blocks is misplaced, none of these blocks is clear and there is a block on top of each of these blocks that is not the block on top of that block in the goal state.

End of exam