Faculteit der Exacte Wetenschappen	Tentamen Ontwerp van Multi-agentsystemen
Vrije Universiteit Amsterdam	22 januari 1999

Opgave	**1**	**2**	**3**	**4**	**bonus**
Punten	20	20	20	40	10

Normering:
Het **tentamencijfer** T is gelijk aan (het totaal behaalde punten voor de tentamenopgaven plus 10 punten bonus) gedeeld door 11.

Het **eindcijfer** voor het hoorcollege Ontwerp van Multi-agentsystemen wordt als volgt berekend:

$$\textbf{Eindcijfer} = (T + H + P) / 3,$$

waarbij
T = tentamencijfer,
H = cijfer huiswerkopgaven,
P = cijfer voor 1 punts practisch werk.

Verzoek:
U wordt vriendelijk verzocht om voor Uw antwoorden gebruik te maken van de speciale **antwoordvellen** voor opgave 1.

Request:
You are kindly requested to make use of the special **antwoordvellen** for Question 1.

U treft aan:
- 4 opgaven
- 3 appendices
- 4 antwoordvellen voor opgave 1

You will find:
4 questions
3 appendices
4 answering sheets for question 1

Note that the text of the questions is translated in **English**, following the Dutch text.

Opgave 1 (20 punten)

Deze opgave bestaat uit twee onderdelen. Motiveer Uw antwoorden.

Verzoek:
U wordt vriendelijk verzocht om voor Uw antwoorden gebruik te maken van de speciale **antwoordvellen** voor opgave 1.

Opgave 1a (10 punten)

In hoofdstuk 1 van de syllabus zijn een aantal primitive agentconcepten geïntroduceerd (zie tabel 1 van de antwoordvellen). In Appendix 1 kun je wat informatie vinden over verzekeringsagent Okke. Analyseer deze informatie aan de hand van de primitieve agentconcepten en vul tabel 1 van de antwoordvellen in. Als de informatie in Appendix 1 naar je zin niet precies genoeg is om de tabel goed te kunnen invullen, dan mag je er zelf informatie bij verzinnen. Denk er aan dat je je antwoorden goed motiveert.

Opgave 1b (10 punten)

In het generieke agentmodel (hoofdstuk 6 van de syllabus) komen een aantal componenten voor. Stel dat je het generieke agentmodel zou gebruiken om een artificiële agent te ontwerpen die Okke's taak zou kunnen overnemen. Gebruik je antwoord op vraag 1a om te beslissen welke van deze componenten je nodig hebt in je ontwerp. Motiveer waarvoor een component nodig is, motiveer ook waarom een component eventueel niet nodig is.

English:

Question 1 (20 points)

This question consists of two parts. Motivate your answers.
Request:
You are kindly requested to make use of the special **antwoordvellen** for Question 1.

Question 1a (10 points)

In chapter 1 of the syllabus a number of primitive agent concepts have been introduced (see table 1 of the answer sheets). In Appendix 1, you can find some information on insurance agent Okke.
Analyse this information according to the primitive agent concepts and fill out table 1 of the answer sheets. In case you feel the information in Appendix 1 is not detailed enough to fill out the table properly, you are allowed to make up additional information. Remember to motivate your answers clearly.

Question 1b (10 points)

In the generic agent model (chapter 6 of the syllabus), there are a number of components. Suppose you would use the generic agent model to design an artificial agent that can take over Okkes task. Use your answer to question 1a to decide which of these components you need in your design. Motivate why a component is needed; in case you left out a component, motivate why this component is not necessary.

Opgave 2 (20 punten)

Bestudeer de partiële specificatie (zie Appendix 2) voor de agent die het maanvehikel Scotty moet besturen. Deze specificatie voldoet prima in de zin dat hiermee Scotty in staat blijkt obstakels te vermijden. Helaas heeft de maan een donkere zijde. Als Scotty aan die donkere kant terecht komt, dan zijn we Scotty kwijt. Scotty rijdt namelijk op lichtcellen. Zonder die energie is communicatie met Scotty onmogelijk en stopt Scotty. De sensoren van Scotty kunnen het verschil tussen licht en donker onderscheiden. Daarmee kan de bestuurder van Scotty waarnemen of het donker is voor, links van, rechts van of achter Scotty.

Gebruik de waarnemingen van de lichtgevoelige sensoren om de kennisbank die in de specificatie in Appendix 2 staat aan te passen zodat Scotty niet alleen obstakels kan vermijden, maar ook de donkere kant van de maan. Schrijf de aangepaste regels van de kennisbank op. (Eventuele andere nodige aanpassingen hoef je niet op te schrijven.)

English:

Question 2 (20 points)

Study the partial specification in Appendix 2. This specification suffices in that this specification enables Scotty to avoid obstacles. Unfortunately, the moon has a dark side. If Scotty ends up on the dark side of the moon, it is lost. Since the power will fail immediately, communications are impossible and Scotty stops. Scotties sensors can also pick up the difference between light and dark: the controller can observe whether it is dark in front of Scotty, to its left, to its right, and behind it.

Use the observations of the light-sensitive sensors to adapt the knowledge base given in the specification in Appendix 2 ensuring that Scotty will both avoid obstacles and the dark side of the moon. Write down the adapted rules of the knowledge base. (If other adaptations of the specification are necessary, you don't have to write those down).

Opgave 3 (20 punten)

Het onderwerp van deze opgave betreft informatietoestanden en redeneren. Bestudeer de partiële specificatie van Appendix 3. Gegeven is de object level public information state S van component mouse_a.

S = [observation_result(at_position(self, p0), pos),
observation_result(at_position(food, p1), pos),
observation_result(at_position(screen, p0), neg)]

Opgave 3a (8 punten)

Geef een informatietoestand S' die S verfijnt en bovendien gesloten en consistent is ten opzichte van de kennisbank van component mouse_a.

Opgave 3b (12 punten)

- Motiveer dat S' een verfijning (refinement) is van S (4 punten).
- Motiveer dat S' gesloten (closed) is ten opzichte van de kennisbank van component mouse_a (4 punten).

Motiveer dat S' consistent is ten opzichte van de kennisbank van component mouse_a (4 punten).

English:

Question 3 (20 points)

This question is about information states and reasoning. Study the partial specification of Appendix 3. This is the object level public information state S of component mouse_a.

S = [observation_result(at_position(self, p0), pos),
observation_result(at_position(food, p1), pos),
observation_result(at_position(screen, p0), neg)]

Question 3a (8 points)

Provide an information state S' that refines S and is also closed and consistent with respect to the knowledge base of component mouse_a.

Question 3b (12 points)

- Motivate that S' is a refinement of S (4 points).
- Motivate that S' is closed with respect to the knowledge base of component mouse_a (4 points).
- Motivate that S' is consistent with respect to the knowledge base of component mouse_a (4 points).

Opgave 4 (40 punten)

Merk op dat de specificatie in Appendix 3 niet uitsluit dat de muis last krijgt van een "gespleten persoonlijkheid". Immers, als de input-informatietoestand als volgt is:

```
[   observation_result(at_position(self, p0), pos),
    observation_result(at_position(food, p1), pos),
    observation_result(at_position(screen, p0), neg),
    observation_result(at_position(food, p0), pos) ]
```

dan zal de muis proberen om tegelijkertijd het voedsel op positie p0 op te eten (hiervoor moet hij op positie p0 blijven) als proberen om naar positie p1 te gaan (om daar later het voedsel op te kunnen eten). Dit probleem treedt op als er meer dan één stuk voedsel in de wereld is. In deze opgave wordt aan U gevraagd om het ontwerp van de muis aan te passen en eventueel uit te breiden zodat de aangepaste muis:

1. slechts één actie tegelijkertijd probeert uit te voeren.
2. als er op twee of meer posities voedsel ligt, dan eet de muis als hij op een positie is waar ook voedsel is (hij blijft dus niet tussen twee posities heen en weer lopen).
3. positie p0 is voor de muis bereikbaar: als er op positie p0 voedsel is en de muis is daar niet, dan gaat de muis naar p0 toe (dit hoeft niet altijd, maar bijvoorbeeld in die wereldtoestanden waarin er maar één stukje voedsel ligt, namelijk op p0).

Voor elke eis waaraan Uw specificatie voldoet krijgt U 10 punten. Daarnaast kunt U nog 10 extra punten krijgen mits Uw specificatie voldoende gemotiveerd is.

English:

Question 4 (40 points)

Note that the specification in Appendix 3 doesn't prevent the mouse from suffering from a "split personality". For when the input information is the following:

```
[   observation_result(at_position(self, p0), pos),
    observation_result(at_position(food, p1), pos),
    observation_result(at_position(screen, p0), neg),
    observation_result(at_position(food, p0), pos) ]
```

the mouse will both attempt to eat the food (for this it has to stay at p0) and go to position p1 (to eat the food there at a later moment). This problem occurs when there is more than one piece of food in the world. You are asked to adjust the design of the mouse and extend it, if necessary, in such a way that the adjusted mouse fulfils these three demands:

1. It tries to perform only one action at a time.
2. In case there is food in more than one position, the mouse eats the food if it is in a position where there is food (it is not running to and fro between two locations).
3. The mouse is able to reach position p0: if there is food at position p0 and the mouse is somewhere else, then the mouse decides to go to p0 (this doesn't always have to happen, but only when for example there is only one piece of food, namely at p0).

You are rewarded with 10 points for every demand your specification fulfils. An additional 10 points is granted when you motivate your design adequately.

Appendix 1

Verzekeringsagent

Verzekeringsagent Okke is iemand die verzekeringen verkoopt aan mensen en bedrijven. Deze persoon gaat bij potentiële klanten op bezoek en bespreekt met de klant waartegen hij/zij zich wil verzekeren. Er zijn verschillende verzekeringsprodukten en verschillende polissen voor die produkten. Okke werkt voor een verzekeringsmaatschappij met een hoofdkantoor waar bepaald wordt welke verzekeringsprodukten deze maatschappij heeft en welke polissen daarvoor gelden. Als hij daar om vraagt, krijgt Okke van het hoofdkantoor alle informatie over de produkten, polissen en klanten die hij nodig heeft om zijn werk te kunnen doen. De verzekeringsagent wordt zelf geacht in de gaten te houden wat de concurrentie aan produkten heeft.

English:

Insurance agent

Insurance agent Okke is selling insurance policies to people and companies. He visits potential customers and discusses with them which risks the client wants covered. There are different insurance products and different policies for those products. Okke works for an insurance company with head-quarters where decisions are made regarding which insurance products are available and which policies there are. If Okke asks for it, he is provided by head-quarters with all information on products, policies and customers he needs to do his job. The insurance agent is supposed to keep track of the products of the competitors by himself.

Appendix 2 Scout control

2.1 Problem Description

A small scout vehicle named Scotty (named after its controller) was dropped on the moon one year ago. In Houston's control room a human controllor (Hannah Scott) is responsible for the control of Scotty. The objective is to obtain pictures of the light side of the moon. Scotty is powered by solar cells which enable it to roam the moon whenever sunlight falls on the solar cells. Scotty has a camera that sends (moving) images to earth continuously during the periods that sunlight falls on Scotties solar cells. Scotty is not autonomous, it receives commands from Houston to determine its course. Scotty drives as long as it has power.

The controller, Hannah Scott, in Houston can give the following commands to Scotty:

- turn left,
- turn right,
- turn back.

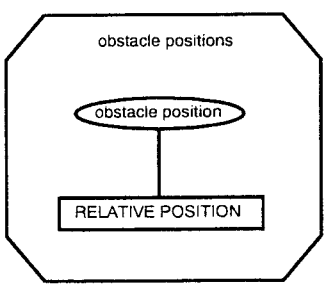
From the moving images the controller receives, Hannah can observe the following:

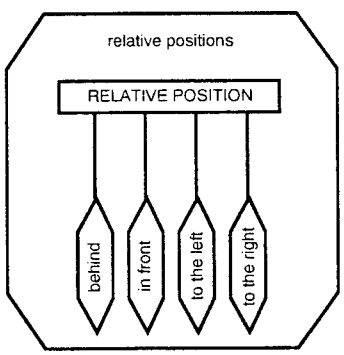
- whether an obstacle is in front of Scotty,
- whether an obstacle is to Scotties left,
- whether an obstacle is to Scotties right,
- whether an obstacle is behind Scotty.

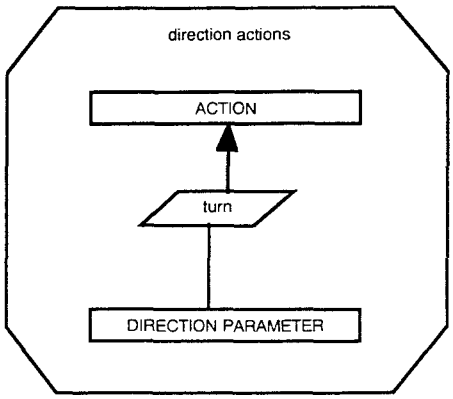
In sections 2.2 and 2.3 of this Appendix a partial specification can be found of the artificial agent that Hannah designed to take over this tedious job.

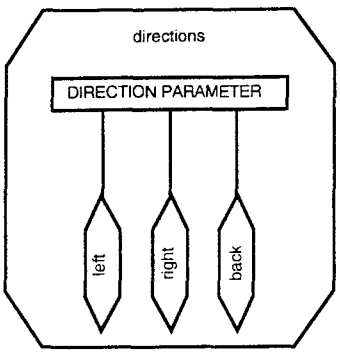
2.2 Informatie typen

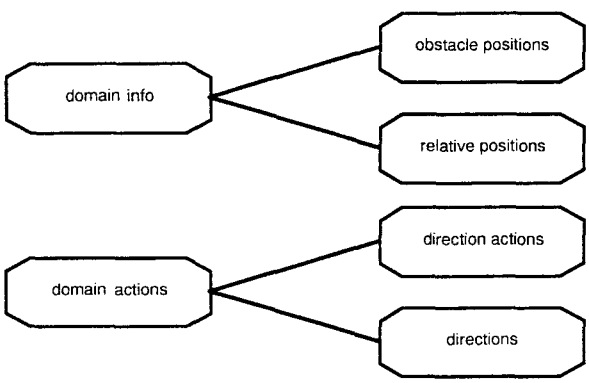
De informatie typen die in het hele systeem gebruikt worden zijn:











information type truth_indication
    sorts SIGN ;
    objects pos,
    neg : SIGN;
end information type

information type observation_results
    sorts INFO_ELEMENT, SIGN ;
    relations observation_result: INFO_ELEMENT * SIGN;
end information type

information type domain_meta_info
    sorts INFO_ELEMENT ;
    meta-descriptions domain_info : INFO_ELEMENT;
end information type

information type observation_result_info
    information types truth_indication,
    observation_results,
    domain_meta_info;
end information type

```
information type actions_to_be_performed
        sorts                                   ACTION ;
        relations       to_be_performed:        ACTION;
end information type

information type action_info
        information types       actions_to_be_performed,
                                domain_actions;
end information type
```

2.3 Fragmenten van specificatie van de component

De component is primitief en wordt hier kort beschreven.

De component agent

De interfaces worden gedefiniëerd door:
input interface: de informatietypen observation_result_info;
output interface: het informatietype action_info;

De targets bij task control focus avoid_undesirable_places zijn:
target(avoid_undesirable_places, to_be_performed(X: ACTION), confirm);

De initial extent is: all-p

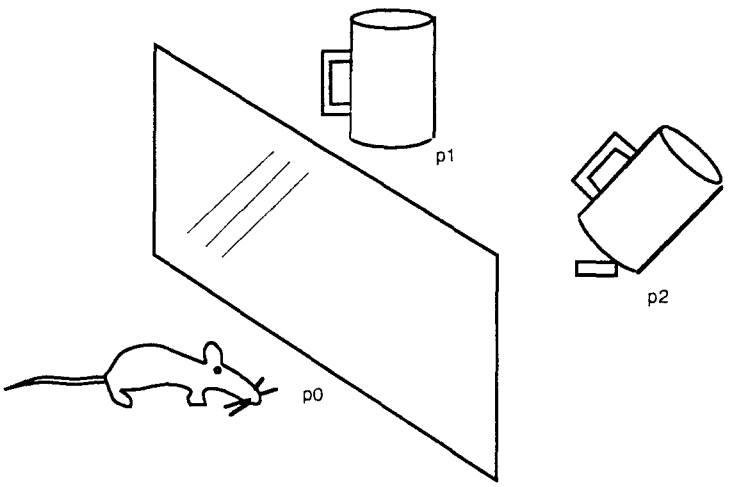
De kennisbank is:

```
if      observation_result(obstacle_position(in_front), pos)
  and   observation_result(obstacle_position(to_the_right), neg)
then    to_be_performed(turn(right));

if      observation_result(obstacle_position(in_front), pos)
  and   observation_result(obstacle_position(to_the_right), pos)
  and   observation_result(obstacle_position(to_the_left), neg)
then    to_be_performed(turn(left));

if      observation_result(obstacle_position(in_front), pos)
  and   observation_result(obstacle_position(to_the_right), pos)
  and   observation_result(obstacle_position(to_the_left), pos)
  and   observation_result(obstacle_position(behind), neg)  /* can be left out for this
domain */
then    to_be_performed(turn(back));
```

Appendix 3 Mouse A

3.1 Problem Description

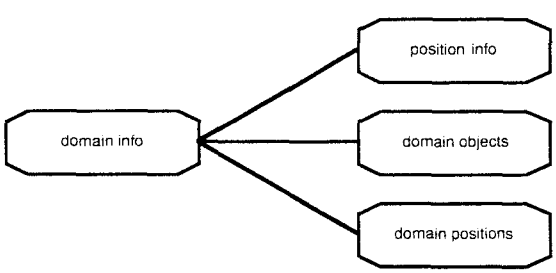
Separated by a transparent screen (a window, at position p0), at each of two positions p1 and p2 a cup (upside down) and/or a piece of food can be placed. At some moment (with variable delay) the screen is raised, and the mouse is free to go to any position. A genuine mouse is known to go to food and eat it.

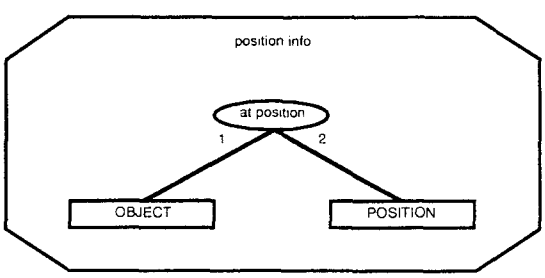


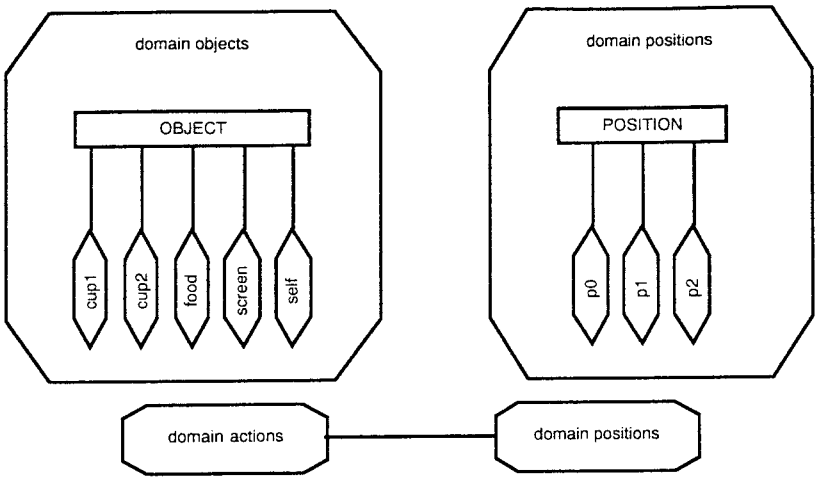
In sections 3.2 and 3.3 of this Appendix a partial specification can be found of the artificial mouse.

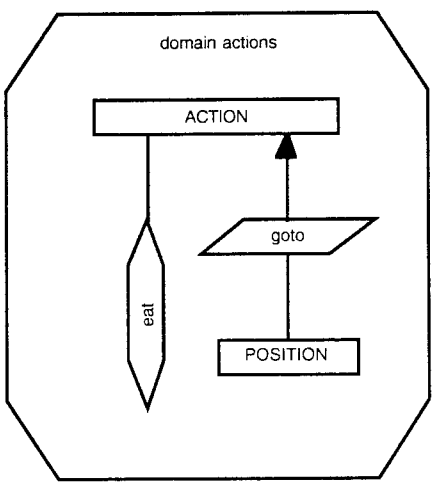
3.2 Informatie typen

De informatie typen die in het hele systeem gebruikt worden zijn:










```
information type truth_indication
        sorts                                   SIGN ;
        objects         pos,
                        neg :                   SIGN;
 end information type

 information type observation_results
        sorts                                   INFO_ELEMENT, SIGN ;
        relations       observation_result:     INFO_ELEMENT * SIGN;
end information type

information type domain_meta_info
        sorts                                   INFO_ELEMENT ;
        meta-descriptions domain_info :         INFO_ELEMENT;
end information type
```

```
information type observation_result_info
        information types      truth_indication,
                               observation_results,
                               domain_meta_info;
end information type

information type actions_to_be_performed
        sorts                                  ACTION ;
        relations      to_be_performed:        ACTION;
end information type

information type action_info
        information types      actions_to_be_performed,
                               domain_actions;
end information type
```

3.3 Fragmenten van specificatie van de component

De component is primitief en wordt hier kort beschreven.

De component mouse_a

De interfaces worden gedefiniëerd door:
input interface: de informatietypen observation_result_info;
output interface: het informatietype action_info;

De targets bij task control focus determine_action zijn:
target(determine_action, to_be_performed(X: ACTION), confirm);

De initial extent is: all-p

De kennisbank is:

```
if      observation_result(at_position(food, P:POSITION), pos)
  and   observation_result(at_position(screen, p0), neg)
  and   observation_result(at_position(self, P:POSITION), neg)
then    to_be_performed(goto(P:POSITION))

if      observation_result(at_position(self, P:POSITION), pos)
  and   observation_result(at_position(food, P:POSITION), pos)
then    to_be_performed(eat)
```

Studentnaam en nummer:_____

Antwoordvellen

Faculteit der Exacte Wetenschappen	Tentamen Ontwerp van Multi-agentsystemen
Vrije Universiteit Amsterdam	22 januari 1999

Studentnaam en nummer:_____

Antwoord 1a

agent concepts and behaviour	Insurance agent Okke
I. External primitive concepts	
A. Interaction with the world	
Passive observation Active observation	
Performing actions	
B. Communication with other agents	
Incoming communication	
Outgoing communication	

Studentnaam en nummer:_____

II. Internal primitive concepts	Insurance agent Okke
A. World Model	
B. Agent Models	
C. Self Model	
D. History	
E. Goals	
F. Plans	

Studentnaam en nummer:_____

agent concepts and behaviour	Insurance agent Okke
G. Group Concepts	
Joint goals	
Joint plans	
Commitments	
Negotiation protocol	
Negotiation strategies	

Studentnaam en nummer:_____

Antwoord 1b

component	*yes / no*	*motivation*
Own Process Control		
Agent Interaction Management		
Maintenance of Agent Information		
World Interaction Management		
Maintenance of World Information		
Agent Specific Task		