Faculteit der Exacte Wetenschappen

Tentamen Ontwerp van Multi-agentsystemen

Vrije Universiteit Amsterdam **26 januari 2001**

Opgave	1	2	3	4
punten	25	20	20	35

Normering:

Het **tentamencijfer T** is gelijk aan het totaal behaalde punten voor de tentamenopgaven gedeeld door 10.

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

Eindcijfer = (T + H + P) / 3,

Waarbij

T = tentamencijfer
H = cijfer huiswerkopgave
P = cijfer voor 1 punts practisch werk.

U treft aan:

4 opgaven
2 appendices

You will find:

4 exercises
2 appendices

Note: All exercises are in English.

Exercise 1 (25 points)

Read the example *Assistant of a Film Director* below and answer the two questions.

Assistant of a Film Director

Q is an assistant of a film director. A big part of her duties is to recruit actors for new roles. She investigates the suitability of an actor with respect to different types of roles, e.g. whether he is capable to play the role of a villain, a hero, etc. In that connection, Q regularly looks for new actors on the Internet and also attends new plays in theaters watching actors performance on astage. She discusses the conclusions of her investigations with the director, and later on she announces to the actors the director's decisions. The actors in their turn ask Q professional advice on their performance and make various requests to Q such as copying scripts, etc. It is her duty to fulfil such requests.

Another part of her obligations is passive monitoring of the filming conditions such as lighting of the stage, sound effects, etc. She discusses her observations with the director, and after receiving his instructions Q informs stage workers and actors what must be improved.

Together with the director and the crew she does her best to produce a film good enough to be nominated for a national film festival. Together with the director she is responsible for making a schedule, so that the film is finished in time.
We asked the director about Q's personal characteristics. Below is the interview with him.

I know Q for five years. She is an energetic person committed to our plans to produce a best film possible. So I need not check her work constantly and give her certain freedom in her decisions.
She started with this job five years ago and was inexperienced. She has learned a lot over the years on this job and became an excellent assistant. She is indispensable in our group. Creative people like us often have difficult features in their characters. I have a bad tempo and never want to go into details of any routine matter and my actors are capricious and demanding. Q often takes initiative to mediate our conflicts and always finds unexpected and effective solutions.
Q is ambitious. In time she wants to be a film director herself, so she studies my work carefully and this year she will negotiate with investors to get money for a small budget movie she is planning to shoot herself.

Question 1A (15 points)

On the base of the description of Q, fill in the tables in Appendix 1.

Question 1B (10 points)

Consider the generic agent model (Chapter 6). It consists of six components:

agent_interaction_management,
world_interaction_management,
maintenance_of_agent_information,
maintenance_of_world_information,
own_process_control,
agent_specific_task.

Which of these components will be useful for the design of a software agent acting in place of Q? Motivate your answer using results from 1A.

Exercise 2 (20 points)

Read the example *Door Safety* below and answer the two questions.

Door Safety
The national bank 'Safe Deposit' has a main safe where most valuable treasures are kept. The door of the main safe is monitored by a software agent consisting of one (primitive) component, which observes the door by means of different special sensors. The incoming information concerns the surrounding temperature, vibration of the door and oxygen concentration.
The temperature parameter has three values: normal, higher_than normal and high; the vibration parameter has two values: no_vibration and vibration; the oxygen parameter has three values: normal, lower_than_normal and low.
On the base of his observations the agent decides what he has to do.

If he sees that temperature and oxygen are normal and there is no vibration, then he decides to take no actions.

If the agent observes that temperature is higher than normal and oxygen is lower than normal and there is no vibration than he decides to alert the bank's security officers.

If the agent observes that oxygen is low or he notices that temperature is high then he decides to alert the local police station.

If the agent sees that the vibration sensor shows vibration then he decides to switch on an electric protection of the door immediately in an attempt to prevent a would-be burglar, and alerts the bank's security officers and the local police station.

Question 2A (10 points):

Formalize the knowledge used in this example. (If you believe that other knowledge rules are relevant, you can add them and give motivation).

Question 2B (10 points):

Give specifications of the information types for the domain specific knowledge that was described in the example.

The relevant chapter for these questions is Chapter 3.

Exercise 3 (20 points)

The subject of this exercise is information states. Read the partial specification of Appendix 2. Consider the following information states S1 and S2 of the component mouse_a.

S1 = [observation_result(at position(self, p0), pos),
observation_result(at position(food, p1), pos),
observation_result(at position(screen, p0), neg)]

S2 = [to_be_performed(goto(p1)), not to_be_performed(goto(p0)]

Question 3A (10 points):

Refine S1 to an information state which is closed and consistent with respect to the knowledge base of the component mouse_a.

Question 3B (10 points):

Consider the following three meta-information states M1, M2, M3:

M1 =
[known(to_be_performed(goto(p1))), true(to_be_performed(goto(p1))),
not false(to_be_performed(goto(p1))),
known(to_be_performed(eat)), not true(to_be_performed(eat)),
false(to_be_performed(eat)),
not known(to_be_performed(goto(p0))), not true(to_be_performed(goto(p0))),
not false(to_be_performed(goto(p0))),
not known(to_be_performed(goto(p2))), not true (to_be_performed(goto(p1))),
not false(to_be_performed(goto(p1)))]

M2 =

[known(to_be_performed(goto(p1))), true(to_be_performed(goto(p1))),

not false(to_be_performed(goto(p1))),

not known(to_be_performed(eat)), not true(to_be_performed(eat)),

false(to_be_performed(eat)),

known(to_be_performed(goto(p0))), not true(to_be_performed(goto(p0))),

false(to_be_performed(goto(p0))),

not known(to_be_performed(goto(p2))), true (to_be_performed(goto(p1))),

not false(to_be_performed(goto(p1)))]

M3 =

[known(to_be_performed(goto(p1))), true(to_be_performed(goto(p1))),

not false(to_be_performed(goto(p1))),

not known(to_be_performed(eat)), not true(to_be_performed(eat)),

not false(to_be_performed(eat)),

known(to_be_performed(goto(p0))), true(to_be_performed(goto(p0))),

false(to_be_performed(goto(p0))),

not known(to_be_performed(goto(p2))), not true (to_be_performed(goto(p1))),

not false(to_be_performed(goto(p1)))]

For each of the following pairs of the information states denote whether or not they are level coherent. For every pair that is not coherent explain why it is so:
(S2, M1), (S2, M2), (S2, M3).

Exercise 4 (35 points).

Read the example *Dog's Health Problem* below and answer the following questions.

Question 4A (25 points):

Determine the levels of process abstraction from the multi-agent perspective and give a process composition. Motivate your answers in the rationale.
Hint: use the generic agent model (Chapter 6) and the generic model for diagnostic reasoning (Chapter 10) and leave out those components that in your opinion have no role in this example.

Question 4B (10 points):

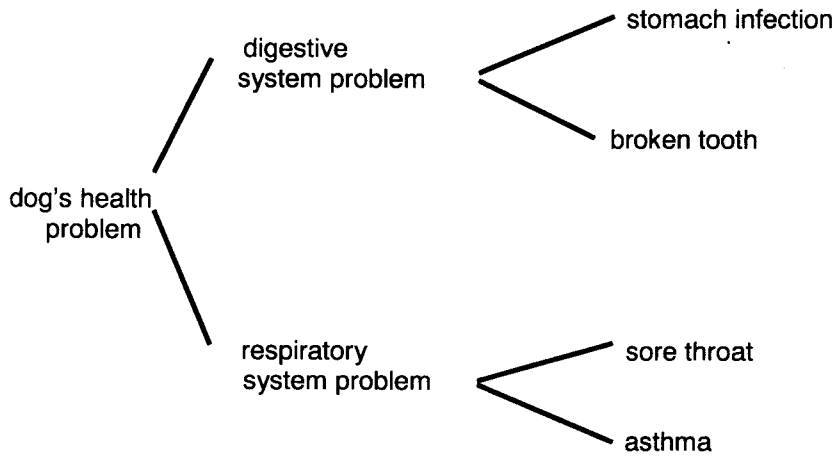
Determine the level of process abstraction from the task perspective.
Give a task delegation. Motivate your answers in the rationale.

Dog's Health Problem

Consider the following situation, which involves two agents, an owner of a dog (Owner) and a vet (Vet). Owner, who is always interested in keeping his dog healthy, observes that his dog is sitting apathetic in a corner, not playing. So Owner identifies this as a problem with the dog's health.

As he is not able to find out himself what the cause of this problem is, he decides to call Vet and ask him to find out the cause of the dog's health problem.

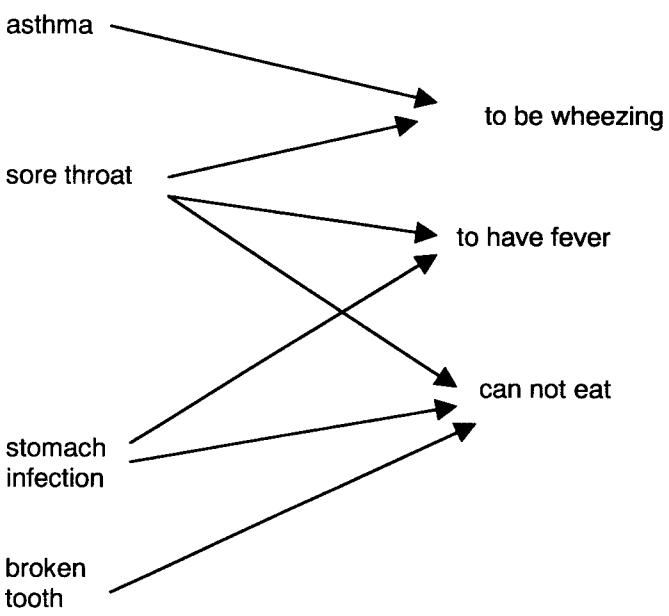
A specific responsibility of Vet is to make a diagnosis of health problems of his animal patients communicated to him by phone. Since Vet has no possibility to observe the dog, he asks Owner to make certain observations and communicate them back to Vet.
Vet has the following hierarchy (taxonomy) of the subproblems of dog's health problem:



The following observations can be made:

- whether or not the dog can not eat
- whether or not the dog is wheezing
- whether or not the dog has fever

The following causal relations between diseases and observations are known to Vet:



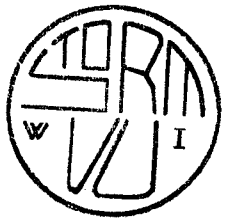
Appendix 1

Vet has also the following (anti-causal) knowledge:

If there is a dog's health problem and the dog can't eat and the dog is not wheezing then there is a digestive system problem;
If there is a dog's health problem and the dog is wheezing then this is a respiratory problem;
If the dog has a respiratory problem and the dog can't eat then the dog has a sore throat;
If the dog has a respiratory problem and can eat then the dog has asthma;
If the dog has a digestive system problem and also fever then it has a stomach infection;
If the dog has a digestive problem and has no fever then it has a broken tooth;
If the dog is not wheezing then this is not a respiratory problem;
If the dog can eat then this is not a digestive problem;
If the dog can't eat and is wheezing then this is not a digestive system problem.

Below you find an example of a possible diagnostic process in finding out the cause of the dog's health problem.

Owner asks Vet (communicates his request to Vet) to find out the cause of the dog's health problem. Vet receives the request communicated by Owner and starts the diagnostic process. Using the above taxonomy, he determines the digestive system problem and respiratory problem as hypotheses to be validated. Considering these problems as focus hypotheses, he determines that two observations should be made: whether the dog can eat and whether the dog is wheezing. Then Vet communicates to Owner the request to make these two observations. After getting this request Owner observes the dog. Suppose he sees that the dog is not wheezing but he can't eat. Owner communicates his observations to Vet.
Using anti-causal knowledge (see the description above) Vet confirms a digestive system problem. Next Vet determines new hypotheses to be validated, namely stomach infection and broken tooth. Vet again communicates to Owner his request to observe whether the dog has fever. After getting the relevant information from Owner that the dog has fever, he evaluates these hypotheses using anti-causal knowledge and finds out that stomach infection is the cause of the dog's health problem. He informs Owner about it.



component Owner

task control foci observations;

input information type target_observation_result_info; /* standard type */
output information type symptoms;

alternative specification user

information links

private link **hypotheses** : object - object
domain assumption_determination
information type assumption_info;
co-domain assumption_evaluation
information type assumption_info;

sort links identity
object links identity
term links identity
atom links
(possible_assumption(HYP: INFO_ELEMENT, S: SIGN),
assumed(HYP: INFO_ELEMENT, S: SIGN)):
<<true,true>, <false,false>,<unknown, unknown>>;
end link

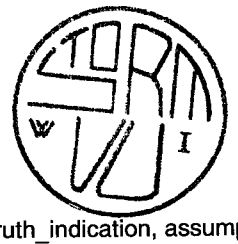
private link **assessments** : object - object
domain assumption_evaluation
information type assumption_info;
co-domain assumption_determination
information type assumption_info;

sort links identity
object links identity
term links identity
atom links
(rejected(HYP: INFO_ELEMENT, S: SIGN),
rejected(HYP: INFO_ELEMENT, S: SIGN)):
<<true, true>, <false, false>>;

(has_been_considered(HYP: INFO_ELEMENT, S: SIGN),
has_been_considered(HYP: INFO_ELEMENT, S: SIGN)):
<<true, true>, <false, false>>;
end link

private link **required_observations** : object - target
domain assumption_evaluation
information type observation_info;
co-domain external_world
information type target_observation_result_info; /* standard type */

sort links (WORLD_INFO_ELEMENT, OA)
object links identity
term links identity
atom links
(to_be_observed(OBS: WORLD_INFO_ELEMENT),
target(observations, OBS: OA, determine)) :
<<true, true>, <unknown, false>, <false,false>>;
end link



information type **assumption_info**
  information types domain_meta_info, truth_indication, assumptions_hypotheses_and_such;
end information type

component assumption_determination

input information types assumption_info, observation_result_info;
output information type assumption_info;

initial kernel information level_2
  assumption(has_been_considered(HYP: INFO_ELEMENT, S: SIGN), neg);

knowledge base assumption_determination_local_kbs
 information types assumption_info, observation_result_info;
contents
/* use as many rules as you like, you may also create additional information types if you like. */

... ...

end knowledge base

component assumption_evaluation

input information types observation_result_info, assumption_info;
output information type observation_info, assumption_info;

knowledge base assumption_evaluation_local_kbs
 information types observation_result_info, assumption_info, observation_info;

contents
if predicted_for(OBS: INFO_ELEMENT, S1: SIGN, HYP: INFO_ELEMENT, S2: SIGN)
then to_be_observed(OBS: INFO_ELEMENT);

if assumed(HYP: INFO_ELEMENT, S: SIGN)
 and predicted_for(OBS: INFO_ELEMENT, pos, HYP: INFO_ELEMENT, S: SIGN)
 and observation_result(OBS: INFO_ELEMENT, neg)
then rejected(HYP: INFO_ELEMENT, S: SIGN)
 and has_been_considered(HYP: INFO_ELEMENT, S: SIGN);

if assumed(HYP: INFO_ELEMENT, S: SIGN)
 and predicted_for(OBS: INFO_ELEMENT, neg, HYP: INFO_ELEMENT, S: SIGN)
 and observation_result(OBS: INFO_ELEMENT, pos)
then rejected(HYP: INFO_ELEMENT, S: SIGN)
 and has_been_considered(HYP: INFO_ELEMENT, S: SIGN);

end knowledge base

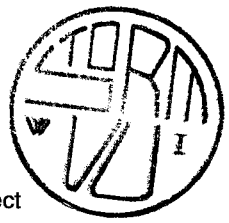
component observation_result_prediction

input information types assumption_info;
output information type assumption_info;

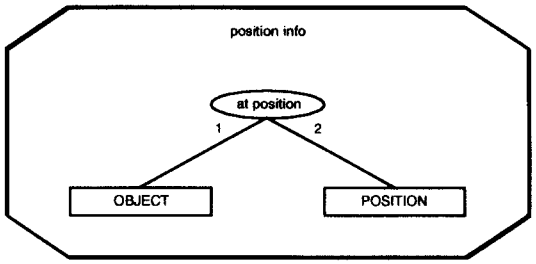
knowledge base observation_result_prediction_local_kbs
 information types assumption_info;

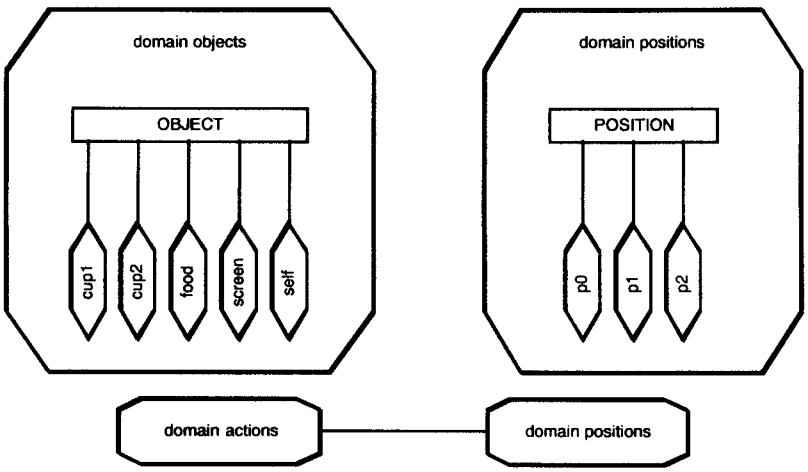
contents
/* use as many rules as you like */

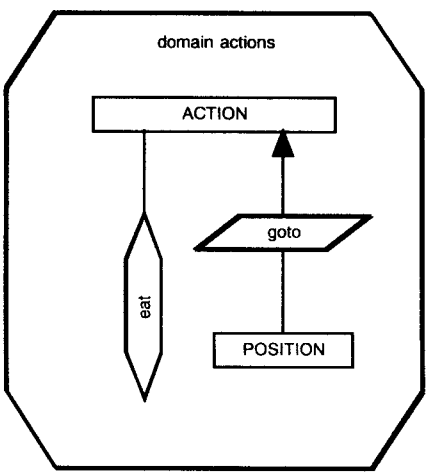
end knowledge base



```
private link observation_results : epistemic - object
domain external_world
        information type epistemic_world_info;        /* standard meta-level */
co-domain assumption_evaluation
        information type observation_result_info; /* object level */

sort links (OA, WORLD_INFO_ELEMENT)
object links identity
term links identity
atom links
        (true(OBS: OA), observation_result(OBS: WORLD_INFO_ELEMENT, pos)) :
                        <<true, true>,<false,false>>;
        (false(OBS: OA), observation_result(OBS: WORLD_INFO_ELEMENT, neg)) :
                        <<true, true>,<false,false>>;
end link


private link assumptions : object - object
domain assumption_determination
        information type assumption_info;
co-domain observation_result_prediction
         information type assumption_info;

sort links identity
object links identity
term links identity
atom links
        (possible_assumption(HYP: INFO_ELEMENT, S: SIGN),
                assumed(HYP: INFO_ELEMENT, S: SIGN)) :
                        <<true, true>, <unknown, false>, <false,false>>;
end link


private link predictions : object - object
domain observation_result_prediction
        information type assumption_info;
co-domain assumption_evaluation
        information type assumption_info;

sort links identity
object links identity
term links identity
atom links
        (predicted_for(OBS: INFO_ELEMENT, S1: SIGN, HYP: INFO_ELEMENT, S2: SIGN),
                predicted_for(OBS: INFO_ELEMENT, S1: SIGN, HYP: INFO_ELEMENT, S2: SIGN))
        :
                        <<true, true>, <unknown, unknown>, <false,false>>;
end link
```


position info
at position
1
2
OBJECT
POSITION


domain objects
OBJECT
cup1
cup2
food
screen
self
domain positions
POSITION
p0
p1
p2
domain actions
domain positions


domain actions
ACTION
goto
eat
POSITION

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

Partial specification of the primitive component mouse_a

input interface
information types observation_result_info;

output interface
information types action_info;

Targets for the task control focus determine_action are:

target(determine_action, to_be_performed(X:ACTION), confirm);

The initial extent is all-p

Knowledge base:

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(goto(eat)) ;

Table 1

External primitive concepts	**has/ has not**	**why**
A. Interaction with the world		
Passive observations		
Active observations		
Performing actions		
B. Communication with other agents		
incoming		
outgoing		

Table 2

Internal primitive concepts	**has/has not**	**why**
A. World model		
B. Agent models		
C. Self model		
D. History		
E. Goals		
F. Plans		
G. Group concepts		
Joint goals		
Joint plans		
Commitments		
Negotiation protocols		
Negotiation strategies		

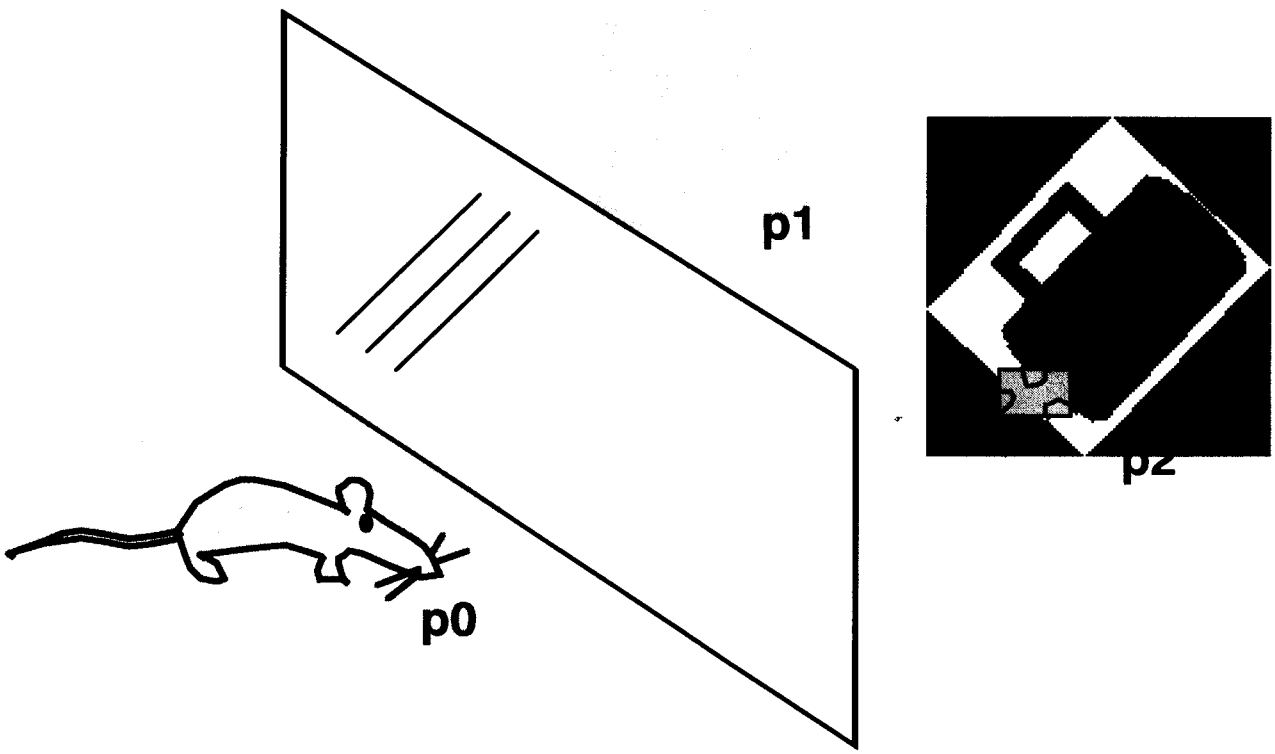
Table 3

Types of behaviour	has/has not	why
Autonomy		
Responsiveness		
Social behaviour		
Own adaptation and learning		

APPENDIX 2.

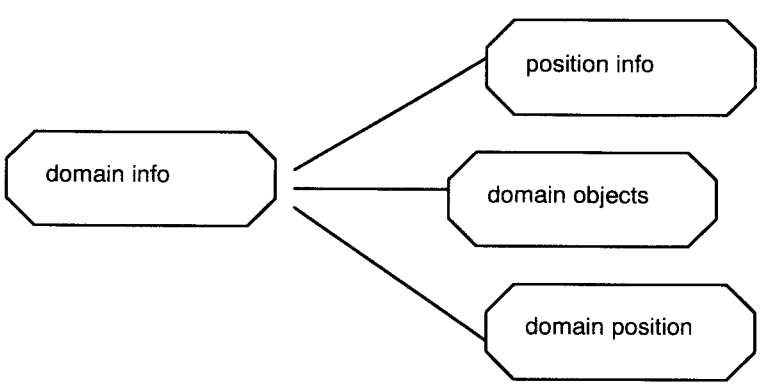
Mouse A.

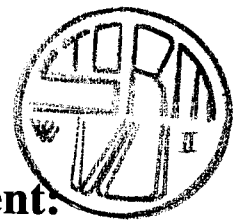
Separated by a transparent screen at position p0, at each of two positions p1 and p2 a cup and/or a piece of food can be placed. At some moment the screen is raised, and the mouse is free to go to any position. Below you can find a partial specification of the artificial mouse.



Information types.

The information types used for the entire system are:





Answer sheet Student: **Year:**

Input (1)	[assumed(a, pos), predicted_for(b, pos, a, pos)]
Output after revision but before reasoning	
Output after reasoning	
Input (2)	[assumed(a, pos), predicted_for(b, pos, a, pos), observation_result(b, neg)]
Output after revision but before reasoning	
Output after reasoning	
Input (3)	[assumed(a, neg), predicted_for(b, neg, a, neg), observation_result(b, neg)]
Output after revision but before reasoning	
Output after reasoning	

Table 1