Multi Agent Systems

Final assignment

For the final assignment you will have to implement a multi-agent system yourselves. You are free to choose a domain. The assignment is done in groups (the same groups as for assignment 2 to 4). You have three weeks for this assignment. In each of the weeks, you have to work on a specific part of the assignment:

- in week 5, you make the conceptual design of your system;
- in week 6, you make a prototype of the implementation in NetLogo;
- in week 7, you make the final NetLogo implementation and a report.

You could discuss your progress with the lab-assistants during the lab-sessions.

Requirements for final assignment

There are a number of requirements for your final system[*]:

- your system consists of multiple agents;
- there is a generic (shared) problem that the agents need to solve collaboratively;
- the agents reason with individual beliefs, desires and intentions (and possibly other mental states);
- the agents act in the environment and thereby influence it;
- the agents communicate with each other.

Grading criteria

The grading will be based on the following aspects:

- the design of the multi-agent system:
  - design of the (cooperation) strategy of the agents (**2 points**)
  - design of communication (ontology, communication protocol) (**1 point**)
  - complexity of the task (**1 point**)
  - creativity and effectiveness of the solution (**1 point**)
- the implementation:
  - correct and consistent application of agent concepts (following the BDI-structure) within the agents (**1 point**)
  - technical quality of the NetLogo implementation (**1 point**)

---

[*] Note that you are welcome (and encouraged) to include more elements, which are not explicitly mentioned here, such as mechanisms for planning, multi-agent coordination or negotiation.

- the report:
  - the research questions (**1 point**)
  - how you use your system to answer the research questions (**1 point**)
  - the completeness of the description of all aspects (**1 point**)

What to hand in?

You have to hand in the intermediate results via Blackboard (although your grade will be based on the final results only). At the end of week 5 (Sunday evening), you submit a document with the conceptual design of your system. You should at least describe:

- the overall task of the agents;
- a (few) research questions you want to answer with your system;
- the characteristics of the environment;
- the communication between the agents;
- the reasoning process of the agents[†].

By the end of week 6, you should submit a first NetLogo implementation of your system (and possibly changes in the setup).

In the final week, you hand in your final NetLogo implementation. In addition, you submit a report in which you describe your system and the design choices you made. In the report you should include:

- a discussion of all aspects that are listed under "Requirements" above (i.e., explaining what are the agents, what problem they need to solve, which mental states they have and how they reason with them, and how they interact with the environment and with each other)[‡];
- a description of a least one simulated scenario;
- a generic conclusion about your model based on your research question(s).

Also describe whether your re-used parts of existing models. The report is expected to be around 5-7 pages.

Please combine the NetLogo model with the report in one .zip-file, and submit this file (as a group) via Blackboard.

[†] To explain how the different beliefs, desires, intentions, and other mental states influence each other, it might be useful to draw a picture to visualize their relationships.

[‡] In general, you can explain all of these things informally (i.e., you don't have to describe the entire NetLogo code in terms of logical rules or something like that). However, in case you use specific mathematical constructs in your code (e.g., like the 'carrying capacity' or 'vision radius' the from earlier assignments), we recommend you to mention them in the report as well.

Some inspiration for application domains

The following ideas might provide you with inspiration for your own system. Your choice is not restricted to these options.

Animal society
There is a group of animals that live together and should do different tasks to survive. For example, they have to find food, bring it to their home, deal with enemies, grow children, etc...

Robot team
A group of robots that have a shared task, e.g. exploring an unknown environment and building a city. For this, the need to understand the environment, collect materials, create new buildings, etc...

Emergency response
A group of first responders (policemen, firefighters, ...) that e.g. have to deal with an incident (e.g., terrorist attack) by building up shared situation awareness, catching bad guys, extinguishing fires, helping wounded people, etc...

Sports team
A group of sports players (e.g., a soccer team) that has to win a game by getting the ball, passing, shooting, etc...

Smart routing in a computer network
A group of components in a computer network that have to send and receive packages without congestion. Agents can be clients that send packages or routers that redirect packages until they reach the final destination. The agents need to exchange information about the traffic to balance the network.

Hunting
A group of predators that hunt different preys. Each predator has its own preference. The predators exchange information in order to decide as a group who is the best prey at that moment (based on criteria like distance, velocity, weight, etc. of the prey but also the hunger of the predator).

Traffic management
Vehicles that have to go from one place to another place, while having several options for their routes. The vehicles should communicate to keep the overall travelling time as short as possible (i.e. to prevent traffic jams).

Warehouse management

Robots that have to operate a warehouse by storing and fetching packages. They need to know were each package is stored and organize their routing and task distribution in an efficient way. They must also avoid collisions with other robots and the shelves.

Smart electricity grid

Households have appliances that consume energy and solar panels that produce energy. The challenge is to balance the production and the consumption of electricity as well as possible. Some appliances can be turned off for some time (e.g. refrigerator), for other appliances the time to turn on can be postponed (e.g. washing machine). The appliances in a neighborhood should communicate with each other to decide when to turn on or off.

Smart cars

Fully automated smart cars in a city, dealing with some cars driven by human beings. The environment consists of crossings, pedestrians and traffic lights. The normal cars are less predictable (they could disrespect red lights, react too late and do not keep enough distance). The smart cars should be able to keep moving in this scenario without getting involved in any accident and should respect the traffic rules all the time. For this, they exchange information about other traffic members.