Multi-Agent Modelling and Verification Environment

Background:

Multi-agent systems are systems where different agents collaborate to complete a joint task. Individual agents have their own goals and yet collaborate in harmony and adapt to their environmental conditions as if they were as single huge agent with specific goals. Agents execute local and simple actions and combine their behavior through different modes of interactions, leading to a notion of emergent (or system-level) behavior. The design and analysis of such behaviors is very difficult but an important endeavor. It is of our interest to design and engineer such behaviors into our advantage.

One of the most prominent examples of multi-agent systems are teams of robots collaborating to achieve a common goal.

We are working on a European project to improve the design and analysis of multi-agent systems. The main idea is to develop a parsimonious but yet an expressive interaction framework to support the design of such systems. Examples of such frameworks include, for example, Reactive Modules (https://www.cis.upenn.edu/~alur/FMSD99.pdf), in which interaction is established based on shared variabls; or π-calculus (http://courses.cs.vt.edu/cs5204/fall09-kafura/Papers/PICalculus/Pi-Calculus-Introduction.pdf), in which interaction is restricted to synchronization through message-passing.

We have designed a mathematical formalism and a basic programming language that support different modes of interactions: (1) agent-to-agent interaction by means of multiple types of message-passing where messages explicitly carry data; and (2) sense and/or manipulate of individual agent's local status and surroundings through shared variables. In our formalism where interaction structures are not only easily reconfigurable but also can be incrementally built at run-time.

Project Description:

The purpose of the project is to extend the tool support for modelling of multi-agent systems.

You will have to get the background information on this type of modelling and analysis as well as understand the particulars of the mathematical model and programming language that we have developed.

Here are several suggested extensions:

1. Extend the programming language with more syntactic options to support more realistic programming.

2. Find good ways for showing simulations that arise from multi-agent system communication and interaction and implement them.
3. Extend our model checking and support to full LTOL (see paper).

Reading Suggestions:

1) Basic programming and model checking support: RCheckTool
2) Mathematical definition of MAS language: http://arxiv.org/abs/1906.10793
3) Model Checking: O. Kupferman, Automata Theory and Model Checking, Ch. 2 in the Handbook of Model Checking https://www.springer.com/gp/book/9783319105741/

Prerequisites:

Prior knowledge in Java, Python, and Javascript

Group size

4-6

Target group

IT, D and DV.

Proposal Authors and Supervisors

Nir Piterman (piterman_at_chelmers_dot_se), Yehia Abd Alrahman (yehiaa_at_chalmers_dot_se)