Multi-Agent Programming Contest 2014
Participation Registration

Michael Ruster

University Koblenz · Landau

Introduction

1. What is the name of your team?
   - Our team is called MAKo. It is an acronym for *Multi-Agents Koblenz*.
2. Who are the members of your team? Please provide names, academic degrees and institutions.
   - All of our team members are from the University Koblenz-Landau.
     - Artur Daudrich,
       Bachelor of Science Informatics.
     - Sergey Dedukh,
       Diploma Applied Mathematics and Informatics.
     - Manuel Mittler,
       Bachelor of Science Business Informatics.
     - Michael Ruster,
       Bachelor of Science Informatics.
     - Michael Sewell,
       Bachelor of Science Computational Visualistics.
     - Yuan Sun,
       Bachelor of Science Information Management and Information Systems.
3. Who is the main-contact? Please also provide an Email address.
   - The main contact is Michael Ruster, mruster@uni-koblenz.de.
4. How much time (man hours) will you have invested (approximately) until the tournament?
   - Every team member will have invested about 300 (wo-)man hours until the tournament. This also includes time spent familiarising with multi agent programming in general as our members were inexperienced in this field.

System Analysis and Design

1. Will you use any existing multi-agent system methodology such as Prometheus, O-MaSE, or Tropos?
   - We did not strictly follow any methodology.
2. Do you plan to distribute your agents over several machines during the competition?

   - We had thought about this due to performance problems we encountered earlier. In the end, we refrained from distributing our agents over several machines.
3. Is the strategic decision making of your team centralized (e.g. on one single agent) or distributed among the agents?
   - We do not have a dedicated leader who commands the team at all times. Instead, agents decide on their own based on various factors and negotiate when necessary. However, our zoning approach has one designated leader per zone group. This agent sends orders to the other agents in her group to form the zone.
4. Describe the communication strategy among agents in your team. Can you estimate the communication complexity in your approach?
   - We found communication to be a very expensive part. Hence, we tried to make agents more independent and reduce the communication this way. Additionally, we have map calculations being done outside of agents on which agents may reason. Due to the agent's independence, the communication is confined to only a few message exchanges e.g. calling an agent for help or negotiating who may travel to a given node. There are more examples but in any case we try to let agents make decisions as quickly as possible.

Software Architecture

1. Which programming language do you plan to use to implement the multi-agent system? (e.g. 2APL, Jason, Jadex, JIAC, Goal, Java, C++, ...)
   - We developed our agents in AgentSpeak(L) and Jason. Map generation and processing is done in Java.
2. Which development platform and tools are you planning to use?
   - We used Eclipse as most team members were familiar with this IDE. Furthermore, we used the Jason mind inspector.
3. Which runtime platform and tools are you planning to use? (e.g. Jade, AgentScape, simply Java, ...)
   - We use Java and Jason's centralised infrastructure.