Multi-Agent Programming Contest 2017
Participation Registration

Introduction

1. What is the name of your team?

   SMART-JaCaMo.

2. Who are the members of your team? Please provide names, academic degrees and institutions.

   Rafael C. Cardoso (PhD Student at PUCRS);
   Tabajara Krausburg (MSc Student at PUCRS);
   Túlio Baségio (PhD Student at PUCRS);
   Débora Engelmann (MSc Student at PUCRS);
   Rafael H. Bordini (PhD, currently Associate Professor at PUCRS);
   Jomi F. Hübner (PhD, currently Associate Professor at UFSC).

3. Who is the main-contact? Please also provide an Email address.

   Rafael C. Cardoso (`rafael_caue@hotmail.com`).

4. How much time (developer hours) will you have invested (approximately) until the tournament?

   Approximately 80 developer hours.

System Analysis and Design

1. Briefly, what is the **main strategy** of the team?

   We start each simulation's round by waiting for jobs. We evaluate any new job to check if its a "good" job. If the job is a mission, we do not evaluate it. If it is a "good"job, or a mission, we decompose the required items into bases and tools, and create contract nets for tasks to buy item bases, to buy tools, to assemble, and to deliver. The contract nets are sent to all available agents, that is, agents who are not yet working on task, or a set of tasks.

2. Will you use any existing multi-agent system **methodology** such as Prometheus, O-MaSE, or Tropos?

   We used Prometheus last year. Thus, our old models were still useful since we repurposed much of the old code, but we did not create any new Prometheus models for our strategies this year.

3. Do you plan to **distribute** your agents on several machines?

   No, we will use only a single machine.

4. Is your solution based on the **centralisation** of coordination/information on a specific agent? Conversely if you plan a decentralised solution, which strategy do you plan to use?

   We are using mostly decentralised solutions. To allocate a job's tasks we are using Contract Net Protocol (CNP). For maintaining a list of available agents (i.e., agents who do not have been allocated a task yet) we are using message passing. And to coordinate job execution (buy, assemble, assist assemble) we are using the Moise organisational model. We do have a centralised structure where agents can share some simple information, such as each agent's role and which tools they can use.

5. Describe the **communication strategy** in the agent team. Can you estimate the communication complexity of your approach?

   Agents only communicate on three occasions: to keep an updated list of available (free) agents, or when the CNP initiator is announcing new tasks, or when the CNP initiator is announcing task winners.

6. Describe the team **coordination strategy** (if any).

   For deciding the tasks that each agent will do in a job, we are using CNP. An agent plays the role of initiator, and all agents play the role of bidders (including the initiator). Agents place bids for tasks (e.g., buying items, buy-

ing tools, assembling items, delivering items) according to their availability (distance to the task, if it can use a specific tool, etc.).

We are also using a centralised structure (a CArtAgO artefact) to store common information. We can use that information in various plans, for example, an agent might need the server name of another in order to execute assist assemble actions, which he can access in the artefact instead of exchanging messages between all agents.

To coordinate job execution we use dynamic scheme creation from the Moise organisational model. Moise enforces that the agents follow the specification, while also allowing them to reason about the goal states in schemes. For example, agents who adopt the mission to provide assistance to assemble, will be able to perceive a change in the assemble goal state once it has concluded, and that they are no longer required.

7. How are the following agent features implemented: *autonomy*, *proactiveness*, *reactiveness*?

   Autonomy: our agents are autonomous by default. JaCaMo uses Jason agents, which are based on the AgentSpeak language. Thus, any goal that contains multiple plans is a choice to be determined by the agent. For example, in our system agents have multiple plans to determine their bid, therefore, we can say that agents determine their bid autonomously.

   Proactiveness: our job strategies and recharge strategies are all proactive. The goal to complete awarded tasks are greater goals which agents continuously work towards. But they can also work on other minor goals at the same time, such as always having enough battery to reach their destination. They do so by plotting a route that guarantees that they will always have battery (if such a route possible).

   Reactiveness: the initiator agent reacts to the announcements of new jobs by decomposing the job's tasks and creating contract nets for them. Agents also react to the "actionId" percept if they are free, by sending a skip action.

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 are implementing the system using the JaCaMo[1] framework.

[1] http://jacamo.sourceforge.net/

2. Which **development** platform and tools are you planning to use?

   We are using Eclipse IDE with JaCaMo plugin.

3. Which **runtime** platform and tools are you planning to use? (e.g. Jade, AgentScape, simply Java, ...)

   Java.

4. Which **algorithms** will be used?

   CNP, some algorithms from our team's last year code, and a few new ones that we developed specifically for this year's scenario iteration.