Multi-Agent Programming Contest 2018

—

The Jason-DTU Team

Jørgen Villadsen, Mads Okholm Bjørn, Andreas Halkjær From,
Thomas Søren Henney, and John Bruntse Larsen

Algorithms, Logic and Graphs Section
Department of Applied Mathematics and Computer Science
Technical University of Denmark
Richard Petersens Plads, Building 324, DK-2800 Kongens Lyngby, Denmark

Abstract. We provide a brief description of the Jason-DTU system, including the overall system design and the tools that we plan to use in the agent contest.

Introduction

The name of our team is Jason-DTU. We participated in the contest in 2009 and 2010 as the Jason-DTU team [1,2], in 2011 and 2012 as the Python-DTU team [3,4], in 2013 and 2014 as the GOAL-DTU team [5], in 2015/2016 as the Python-DTU team [6] and in 2017 as the Jason-DTU team [7].

The members of the team are as follows:

- Jørgen Villadsen, PhD
- Mads Okholm Bjørn, MSc student
- Andreas Halkjær From, MSc student
- Thomas Søren Henney, MSc student
- John Bruntse Larsen, PhD student

We are affiliated with DTU Compute (short for Department of Applied Mathematics and Computer Science, Technical University of Denmark (DTU) and located in the greater Copenhagen area).

The main contact is associate professor Jørgen Villadsen, DTU Compute, email: `jovi@dtu.dk`

We expect that we will have invested approximately 300 man hours when the tournament starts.

System Analysis and Design

The main strategy of the team is to use problem decomposition to efficiently solve subtasks. The subtasks are solved using hierarchical planning, dividing subtasks into even smaller tasks. Each agent should be able to solve as much as possible by itself, coordinating with other agents only when necessary.

No existing multi-agent system methodology has been utilized for the development.

The agents will not be distributed on several machines.

The agents' information is centralized using artifacts provided by the CArtAgO framework. However, the coordination is decentralized using a Contract Net Protocol (CNP) to distribute work. The CNP is implemented using CArtAgO as well, allowing for dynamic creation and announcement of tasks.

The team's communication strategy involves using predefined protocols to solve more complex tasks which requires agent coordination. The agents coordinate their behavior through Jason's communication tools. The communication complexity is $O(n)$ where n is the number of agents involved.

The team's coordination strategy is based on dividing tasks into subtasks, which preferably should be independently solvable by the agents. However, when a task requires agent coordination, help is requested using the CNP, while coordination is handled by utilizing the communication protocols.

The agents are autonomous by autonomously deciding which tasks to complete and which actions to perform in order to solve these. They are proactive by constantly pursuing goals and reactive by reacting to changes in the environment, such as item availability and their charge.

Software Architecture

We have used the Jason and Java programming languages.

The multi-agent system is implemented using the Java development platform.

The runtime platform consists of the Jason programming language integrated with the CArtAgO framework.

We do not use any particular algorithms.

Evaluation

At registration time we cannot estimate how much of our agent team we can recycle from last time.

The strengths of our team are that we are experienced in artificial intelligence and multi-agent systems and that we have a solid theoretical computer science background.

The weaknesses of our team are that we have limited time for analysis, design and programming and that the main developers do not know the Jason programming language in advance.

We hope that our choice of the Jason programming language will pay off.

References

1. Niklas Skamriis Boss, Andreas Schmidt Jensen, and Jørgen Villadsen. *Building Multi-Agent Systems Using Jason.* Annals of Mathematics and Artificial Intelligence, 59:373-388, Springer 2010.
2. Steen Vester, Niklas Skamriis Boss, Andreas Schmidt Jensen, and Jørgen Villadsen. *Improving Multi-Agent Systems Using Jason.* Annals of Mathematics and Artificial Intelligence, 61:297-307, Springer 2011.
3. Mikko Berggren Ettienne, Steen Vester, and Jørgen Villadsen. *Implementing a Multi-Agent System in Python with an Auction-Based Agreement Approach.* Lecture Notes in Computer Science, 7217:185-196, Springer 2012.
4. Jørgen Villadsen, Andreas Schmidt Jensen, Mikko Berggren Ettienne, Steen Vester, Kenneth Balsiger Andersen, and Andreas Frøsig. *Reimplementing a Multi-Agent System in Python.* Lecture Notes in Computer Science, 7837:205-216, Springer 2013.
5. Jørgen Villadsen, Andreas Schmidt Jensen, Nicolai Christian Christensen, Andreas Viktor Hess, Jannick Boese Johnsen, Øyvind Grønland Woller, and Philip Bratt Ørum. *Engineering a Multi-Agent System in GOAL.* Lecture Notes in Computer Science, 8245:329-338, Springer 2013.
6. Jørgen Villadsen, Andreas Halkjær From, Salvador Jacobi and Nikolaj Nøkkentved Larsen. *Multi-Agent Programming Contest 2016 — The Python-DTU Team.* International Journal of Agent-Oriented Software Engineering 6(1):86-100 2018.
7. Jørgen Villadsen, Oliver Fleckenstein, Helge Hatteland and John Bruntse Larsen. *Engineering a Multi-Agent System in Jason and CArtAgO.* Annals of Mathematics and Artificial Intelligence (Online First) 2018.

Acknowledgements

Thanks to Oliver Fleckenstein and Helge Hatteland for insights based on the Jason-DTU team last year.

Further details about the previous DTU teams are available here:

`http://people.compute.dtu.dk/jovi/MAS/`