

# Multi-Agent Programming Contest 2019

---

## The GOAL-DTU Team

Jørgen Villadsen, Alexander Birch Jensen, and Andreas Halkjær From

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 GOAL-DTU system, including the overall system design and the tools that we plan to use in the agent contest.

## General

### Introduction

The name of our team is GOAL-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 and 2018 as the Jason-DTU team [7].

The members of the team are as follows:

- Jørgen Villadsen, PhD
- Alexander Birch Jensen, PhD student
- Andreas Halkjær From, MSc 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](mailto:jovi@dtu.dk)

We expect that we will have invested approximately 100 man hours when the tournament starts.

## Specifics

### System Analysis and Design

1. What is the **main strategy** of the team? To collect all blocks possible, preferably ones needed for tasks, then incorporate to build the patterns. The relative positioning of the agents is exploited to (over time) gain as much positional knowledge as possible.
2. Do you use any existing MAS **methodology**, e.g. Prometheus, O-MaSE, or Tropos? The methodology is tailored to the GOAL loop (no durative / composed planning).
3. Do you plan to **distribute** your agents on several machines? 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? Currently, no optimization is used for shared knowledge. Each agent has its own beliefs.
5. Describe the **communication strategy** of the agent team. Communicate to determine when tasks can be completed, to combine patterns and to share positional knowledge.
6. Describe the team **coordination strategy** (if any). Currently no strategy exists.
7. How are the following agent features implemented: *autonomy*: the agent responds based on its own beliefs, *proactiveness*: the agent will explore the world to find objects and collect blocks before tasks appear, *reactiveness*: the agent (currently) never assumes obstacles are constant; the agents do not try to pick up blocks of other agents.

### Software Architecture

1. Which **programming language** did you use to implement the multi-agent system? (e.g. 2APL, Jason, Jadex, JIAC, Goal, Java, C++, ...) GOAL.
2. Which **development** platform and tools are you using? Eclipse for GOAL.
3. Which **runtime** platform and tools are you using? (e.g. Jade, AgentScape, simply Java, ...) Only Java (through GOAL).
4. Which **algorithms** will be used? To be determined.

### Evaluation

1. What are strengths of your (agent) team? Flexibility by the GOAL approach.
2. What are the (known) weaknesses of your (agent) team? Communication and pathing.
3. Do you think your choice of programming language and/or frameworks paid off? Time will tell. GOAL was chosen for scientific purposes.
4. If you used anything MAS-related, please specify what you think the benefits were (regarding implementation effort, outcome, etc.). Nothing to add.

## 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, 84:57-74, Springer 2018.

Further details about the previous DTU teams are available here:

<http://people.compute.dtu.dk/jovi/MAS/>