Multi-Agent Programming Contest 2011 Techinical Report Template

Abstract. Please follow the given template structure for your submission. For each section, we are proposing some questions that the reader of the paper should be able to answer after reading it. (Please note that the technical report format is not like the "form" used in the registration template. A *paper* style should be used.)

If you encounter any problems with some of the questions please do not hesitate to request clarification on the mailing list.

Max number of pages: 10

Deadline: **20/10/2011**

Format: LNCS

`http://www.springer.com/computer/lncs?SGWID=0-164-6-793341-0`

Please submit your contribution via

`https://www.easychair.org/conferences/?conf=mapc11`

(menu TechReport).

1 Introduction

1. What was the motivation to participate in the contest?
2. What is the (brief) history of the team? (MAS course project, thesis evaluation, ...)
3. What is the name of your team?
4. How many developers and designers did you have? At what level of education are your team members?
5. From which field of research do you come from? Which work is related?

2 System Analysis and Design

1. If some multi-agent system methodology such as Prometheus, O-MaSE, or Tropos was used, how did you use it? If you did not what were the reasons?
2. Is the 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?
3. What is the communication strategy and how complex is it?
4. How are the following agent features considered/implemented: *autonomy*, *proactiveness*, *reactiveness*?
5. Is the team a truly **multi**-agent system or rather a centralised system in disguise?
6. How much time (man hours) have you invested (approximately) for implementing your team?
7. Did you discuss the design and strategies of your agent team with other developers? To which extent did your test your agents playing with other teams?

3   Software Architecture

1. Which programming language did you use to implement the multi-agent system?
2. Did you use multi-agent programming languages? Why or why not to use a multi-agent programming language?
3. How have you mapped the designed architecture (both multi-agent and individual agent architectures) to programming codes, i.e., how did you implement specific agent-oriented concepts and designed artifacts using the programming language?
4. Which development platforms and tools are used? How much time did you invest in learning those?
5. Which runtime platforms and tools (e.g. Jade, AgentScape, simply Java, ...) are used? How much time did you invest in learning those?
6. What features were missing in your language choice that would have facilitated your development task?
7. What features of your programming language has simplified your development task?
8. Which algorithms are used/implemented?
9. How did you distribute the agents on several machines? And if you did not please justify why.
10. To which extend is the reasoning of your agents synchronized with the receive-percepts/send-action cycle?
11. What part of the development was most difficult/complex? What kind of problems have you found and how are they solved?
12. How many lines of code did you write for your software?

4   Strategies, Details and Statistics

1. What is the main strategy of your team?
2. How does the overall team work together? (coordination, information sharing, ...)
3. How do your agents analyze the topology of the map? And how do they exploit their findings?
4. How do your agents communicate with the server?
5. How do you implement the roles of the agents? Which strategies do the different roles implement?
6. How do you find good zones? How do you estimate the value of zones?
7. How do you conquer zones? How do you defend zones if attacked? Do you attack zones?
8. Can your agents change their behavior during runtime? If so, what triggers the changes?
9. What algorithm(s) do you use for agent path planning?
10. How do you make use of the buying-mechanism?
11. How important are achievements for your overall strategy?

12. Do your agents have an explicit mental state?
13. How do your agents communicate? And what do they communicate?
14. How do you organize your agents? Do you use e.g. hierarchies? Is your organization implicit or explicit?
15. Is most of your agents' behavior emergent on and individual and team level?
16. If your agents perform some planning, how many steps do they plan ahead?

5 Conclusion

1. What have you learned from the participation in the contest?
2. Which are the strong and weak points of the team?
3. How suitable was the chosen programming language, methodology, tools, and algorithms?
4. What can be improved in the context for next year?
5. Why did your team perform as it did? Why did the other teams perform better/worse than you did?
6. Which other research fields might be interested in the Multi-Agent Programming Contest?
7. How can the current scenario be optimized? How would those optimization pay off?

Short Answers

Please provide short answers to all the questions in a separate section. This does not count for the 10 pages limit.