Exercício Prático 2: MAPC Agents on Mars

Alan Rafael Fachini[1], Lucas Nascimento[1], and Márcio F. Stabile Jr.[2]

[1] PCS-EPUSP
[2] IME-USP

Resumo Este relatório descreve a implementação de um time usando a abordagem de sistema multi-agente para a competição no senário "Agents on Mars" desenvolvida pelo "Multi-Agent Programming Contest" (MAPC). Neste senários os agentes devem encontrar as melhores zonas com os maiores pesos nos nós do grafo. A implementação apresentada usou como base o sistema "LTI-USP Team" desenvolvida por Franco e Sichman [1] para a competição de 2013, e testes com estratégias diferentes para a organização dos times foram realizados a fim de tentar identificar estratégias mais eficientes.

1 Introdução

TODO

1. What was the motivation to participate in the contest?
   O MAPC tem como um de seus principais objetivos estimular a pesquisa na área de sistemas multi-agentes. O objetivo deste trabalho é então propor uma estratégia e gerar um time de agentes que participe da competição se utilizando dela.
2. What is the (brief) history of the team? (MAS course project, thesis evaluation, . . .)
   O time foi então criado não só para o fim de pesquisa, como também para participar do campeonato interno da disciplina de Sistemas Multi-Agentes (PCS 5703) da Universidade de São Paulo de forma a cumprir parte dos requisitos necessários para obtenção dos créditos da disciplina.
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?
   O time tem como nome PCS-ALM e é composto de três integrantes:
   – Alan Rafael Fachini - Cursando Mestrado em Engenharia da Computação (PCS-EPUSP)
   – Lucas Nascimento - Cursando Mestrado em Engenharia da Computação (PCS-EPUSP)
   – Márcio F. Stabile Jr. - Cursando Mestrado em Ciência da Computação (IME-USP)
5. From which field of research do you come from? Which work is related?
   Os integrantes tem ligação com pesquisas na área de Inteligência Artificial

2 Projeto e Análise do Sistema

1. Did you use multi-agent programming languages? Please justify your answer.
   Para o desenvolvimento dos agentes foi utilizada a linguagem Jason, implementação do AgentSpeak que está contida no arcabouço JaCaMo que permite criar a implementação do agente e também o esquema organizacional.
2. If some multi-agent system methodology such as Prometheus, O-MaSE, or Tropos was used, how did you use it? If you did not, please justify.
   Não foram utilizadas metodologias de desenvolvimento primeiramente pelo fato dos agentes terem sido desenvolvidos com base em agentes já existentes e também pelo baixo limite de tempo que não permitiu aos membros estudar a fundo metodologias para o desenvolvimento.
3. 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?
4. What is the communication strategy and how complex is it?
5. How are the following agent features considered/implemented: *autonomy*, *proactiveness*, *reactiveness*?
6. Is the team a truly **multi-**agent system or rather a centralised system in disguise?
7. How much time (man hours) have you invested (approximately) for implementing your team?
8. Did you discuss the design and strategies of you agent team with other developers? To which extend did your test your agents playing with other teams.
9. What data structures are shared among the agents, and which are private of each agent?

3 Software Architecture

1. Which programming language did you use to implement the multi-agent system?
2. 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?
3. Which development platforms and tools are used? How much time did you invest in learning those?
4. Which runtime platforms and tools (e.g. Jade, AgentScape, simply Java, . . .) are used? How much time did you invest in learning those?
5. What features were missing in your language choice that would have facilitated your development task?
6. Which algorithms are used/implemented?
7. How did you distribute the agents on several machines? And if you did not please justify why.

8. Do your agents perform any reasoning tasks while waiting for responses from the server, or is the reasoning synchronized with the receive-percepts/send-action cycle?
9. What part of the development was most difficult/complex? What kind of problems have you found and how are they solved?
10. 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 you agents' behavior emergent on and individual and team level?
16. If you agents perform some planning, how many steps do they plan ahead.
17. If you have a perceive-think-act cycle, how is it synchronized with the server?

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. Please use the following style for this section:

```
\newpage
\section*{Short Answers}
\appendix
\section{Introduction}
\begin{enumerate}
\item What was the motivation to participate in the contest?
\item[A:] Our motiviation was ...
\item What is the (brief) history of the team?
(MAS course project, thesis evaluation, $\ldots$)
\item[A:] In 2006...
\end{enumerate}
```

Please note: The `A:` stands for ”‘Answer”’.

Referências

1. Mariana Ramos Franco and Jaime Simão Sichman. Improving the lti-usp team: A new jacamo based mas for the mapc 2013. In *Engineering Multi-Agent Systems*, pages 339–348. Springer, 2013.