

Multi-Agent System in Data Center environment

Douglas Trajano

Pontifical Catholic University of Rio Grande do Sul - PUCRS
School of Technology, Porto Alegre, Brazil
douglas.trajano@edu.pucrs.br

Abstract

In this paper, we introduce a multi-agent system in a data center environment. The system is composed of different types of agents, each of which has its artifact, and the system is designed to solve all issues in the data center environment. The agents walk around the data center searching for issues, when they find an issue, they will try to solve it. The system is designed to solve all issues in the data center environment.

1 Introduction

Nowadays, IT operations are crucial for business continuity. Information systems are used by companies in different processes and products, all these services are implemented in data centers. Critical incidents in data centers may impact the usability of these services and it needs to be handled in a timely fashion by the engineering team to avoid financial losses or affect their end-users.

A data center is a building composed of computer systems, storage systems, infrastructure for power supply, telecommunications, environmental controls (e.g., air conditioning, fire suppression), and many other associated components. The complexity of these resources lies in the fact that all these components need to work together, if one of these components has an issue, it can affect other components. Engineers need to solve these issues in a timely fashion.

In this work, we introduce a multi-agent system in a data center environment. The environment simulates a data center plant with several racks, each rack has several servers. When a server is facing an issue, it will be marked as "red", and the agents will try to solve the issue. The system is designed to solve all issues in the data center environment.

This work is organized as follows. Section 2 is about the related work and the background of the problem. Section 3 explains our approach to solve the problem. Section 4 describes our experiments and the results. Finally, we conclude with discussion and future directions in Section 5.

2 Related work and background

Modern software applications have to deal with an increasing level of autonomy of interconnected software systems. Multi-agent systems are a paradigm for modeling and engineering complex systems. (Boissier et al. 2020)

As the name says, Multi-Agent Systems (MAS) are systems made up of multiple agents. It has the capacity to build and understand a wide range of artificial social systems and can be applied in several different areas. (Wooldridge 2009)

JaCaMo (Boissier et al. 2013) is a multi-agent systems development platform that enables integration of three multi-agent programming dimensions: agents, organisations, and environment. A JaCaMo system, as shown in Figure 2.1, consists of the following platforms: Jason (Bordini, Hübner, and Wooldridge 2007) for agent development, CArtaGo (Bordini et al. 2009) for environment programming, and Moise (Hubner, Sichman, and Boissier 2007) for programming organisations. Therefore, JaCaMo integrates these three platforms for a uniform and consistent programming model, with the goal of simplifying the combination of these dimensions when programming multi-agent systems (Boissier et al. 2013).

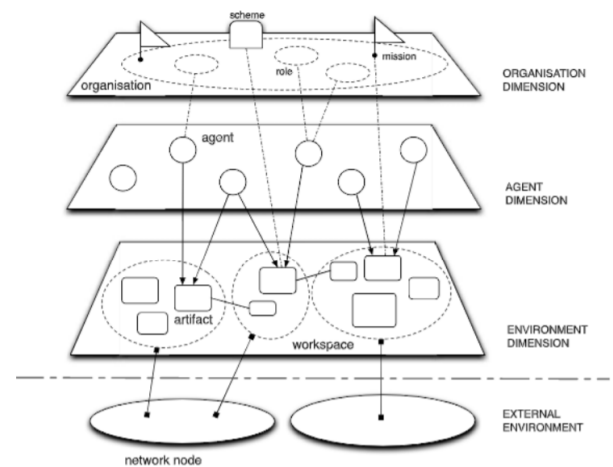


Figure 1: Overview of a JaCaMo Multi-Agent System (Boissier et al. 2013)

3 Approach

Our approach is based on the following three steps:

3.1 Environment

Our environment simulates a data center plant. It has several racks, and each rack has several servers. The cells with servers are marked as "gray" and the agents do not move to these cells until the servers are broken. Cells with hardware issues are marked as "red" and cells with software issues are marked as "orange". When an agent tries to solve an issue, it will move to these cells. Free cells are marked as "white" and the agents can move to these cells. In Figure 3.1, we can see the map that represents the environment.

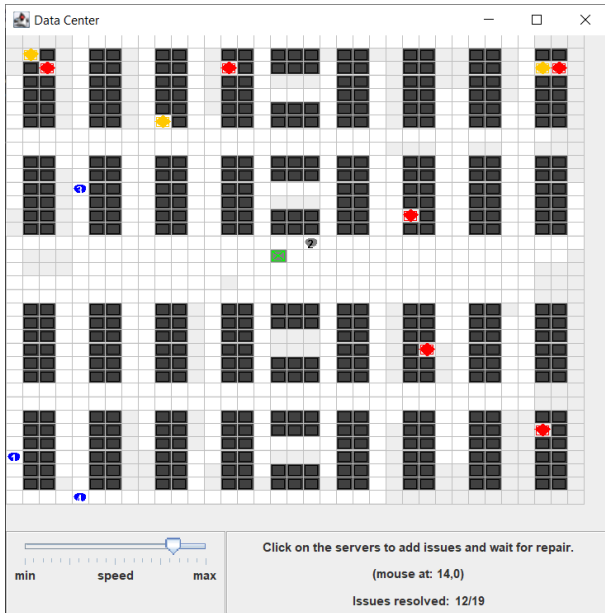


Figure 2: Data Center - Environment

3.2 Problem

The problem that our agents are trying to solve is to find all the issues in the data center environment. For that, agents will walk around the environment and try to find issues. When an agent finds an issue, it will try to solve it. If all the issues are solved, the agents still walk around searching for new issues as we can interact with the environment in real-time adding new issues.

3.3 Agents

Basically, we have two types of agents, the manager, and the technicians. For technician agents we will have different goals and beliefs.

The **manager** is responsible for coordinating the workers, it isn't visible in the environment, but it will manage the team scores. We introduced the concept of teams. Currently, each team has only one agent, but we can add more agents for these teams. In the end, the manager will say the team that has the highest score.

The **technician** agents represent possible roles in a data center such as "facilities technician", "hardware engineer", "network engineer" and "system administrator". They will try to solve the issues in the environment. When they find an issue, they will try to solve it. If they succeed, they will get a score. Software issues only can be solved by the System Admin agent and hardware issues can be solved by the other agents.

We also have the concept of the team. In our experiments, we added one agent per team, but we can add more agents in each team. The manager calculates the score of the teams.

4 Experiments

We developed one scenario to test our Multi-Agent System. This scenario has 410 servers in 400 racks. It starts with 3 software issues and 19 hardware issues. We also can add more issues by clicking in a server in the world interface, when we add an issue manually, the issue type (hardware or software) will be defined randomly.

Our agents start walking around the world and try to find and solve the issues. When all issues are solved, the agents still have to walk around the world because new issues can occur.

5 Discussion, Conclusion, and Future Works

We can conclude that we can introduce a Multi-Agent System that controls autonomous agents in a data center world.

As we discussed in the Introduction section, data centers are complex resources. So, for future work, we want to specify more types of issues that can occur. We also can define issues that requests more than one agent role to fix it. Another possibility is define if the agent should return to the command center or not based on the issue type. We can add more components to the infrastructure such as network devices, air conditioning, and so on.

References

- Boissier, O.; Bordini, R. H.; Hübner, J. F.; Ricci, A.; and Santi, A. 2013. Multi-agent oriented programming with jacamo. *Science of Computer Programming* 78(6):747–761.
- Boissier, O.; Bordini, R. H.; Hubner, J.; and Ricci, A. 2020. *Multi-agent oriented programming: programming multi-agent systems using JaCaMo*. MIT Press.
- Bordini, R. H.; Dastani, M.; Dix, J.; and Seghrouchni, A. E. F. 2009. *Multi-agent programming*. Springer.
- Bordini, R. H.; Hübner, J. F.; and Wooldridge, M. 2007. *Programming multi-agent systems in AgentSpeak using Jason*, volume 8. John Wiley & Sons.
- Hubner, J. F.; Sichman, J. S.; and Boissier, O. 2007. Developing organised multiagent systems using the moise+ model: programming issues at the system and agent levels. *International Journal of Agent-Oriented Software Engineering* 1(3-4):370–395.
- Wooldridge, M. 2009. *An introduction to multiagent systems*. John Wiley & sons.